TABLE OF CONTENTS

1.	Introduction	1	1/A7
2.	Philosophy of BIRP Design.	2	1/A8
3.	BIRP Data File Structures.	3	1/A9
	3.1 The PARAMETER.BRP Files	4	1/A10
	3.1.1 The Header Area.	4	1/A10
	3.2.2 The Data Area.	7	1/A13
	3.2 The BIRP Bit Map.	8	1/A14
	3.3 The Index File in BIRP.	8	1/A14
	3.4 The Scan Control Table in BIRP.	9	1/B1
	3.5 The BIRP.BRP File	10	1/B2
4.	Main Program - Overview and Overlay Structure.	10	1/B2
5.	Describe	12	1/B4
6.	Search	12	1/B4
	6.1 Parameter Input	13	1/B5
	6.2 Parameter Search.	15	1/B7
	6.2.1 Bitwise Scan	15	1/B7
	6.2.2 Sorted List and Range Scan	16	1/B8
	6.2.3 Random List and Range Scan	17	1/B9
	6.2.4 Area Scan.	18	1/B10
7.	Print.	20	1/B12
	7.1 Add	20	1/B12
	7.2 Clear	21	1/B13
	7.3 Copy.	21	1/B13
	7.4 Delete.	21	1/B13
	7.5 Describe.	22	1/B14
	7.6 Dump.	22	1/B14
	7.7 Micro	22	1/B14
	7.8 Next.	24	1/C2
	7.9 Previous.	24	1/C2
	7.10 Print	25	1/C3
	7.11 Start	25	1/C3
8.	Save	26	1/C4
9.	Restore and Backup	27	1/C5

10.	Examples of BIRP Sessions with Viking Orbiter Library. . .	27	1/C5
	10.1 Log-on and HELP.	27	1/C5
	10.2 DESCRIBE Viking Orbiter Library.	29	1/C7
	10.3 VO Incidence, Emission, Phase Angles similar to Viking Lander Pictures	30	1/C8
	10.4 Box Area Search of Lunae Planum.	31	1/C9
	10.5 Highest Resolution, Clearest Pictures of Mars. . . .	34	1/C12
	10.6 Low Sun Angle Pictures of Tharsis Plains	36	1/C14
	10.7 Limb Shots with Red Filter During Dust Storms. . . .	39	1/D3
	10.8 Seasonal Variations in a Crater Streak in the Elysium Quadrangle	40	1/D4
11.	Autogen.	43	1/D7
	11.1 System Considerations.	43	1/D7
	11.2 Data Structures for Autogen.	44	1/D8
	11.3 Running Autogen.	45	1/D9
	11.4 Program Flow Through Autogen	47	1/D11
12.	Create	49	1/D13
	Appendix 1 Building BIRP on Your System	51	1/E1
	Appendix II Description of BIRP Routines	54	1/E4
	Appendix III An Abbreviated Guide to BIRP Commands.	60	1/E10

L 2 1 1980

NAS 1.26: 3299

COMPLETED

ORIGINAL

NASA Contractor Report 3299

BIRP: Software for Interactive Search and Retrieval of Image Engineering Data

Raymond E. Arvidson, Lawrence K. Bolef,
Edward A. Guinness, and Peter Norberg

GRANT NSG-7087
JULY 1980

NASA

NASA Contractor Report 3299

BIRP: Software for Interactive Search and Retrieval of Image Engineering Data

Raymond E. Arvidson, Lawrence K. Bolef,
Edward A. Guinness, and Peter Norberg
McDonnell Center for the Space Sciences
Washington University
St. Louis, Missouri

Prepared for
NASA Headquarters
under Grant NSG-7087

National Aeronautics
and Space Administration

**Scientific and Technical
Information Office**

1980

TABLE OF CONTENTS

1. Introduction 1

2. Philosophy of BIRP Design. 2

3. BIRP Data File Structures. 3

3.1 The PARAMETER.BRP Files 4
3.1.1 The Header Area. 4
3.2.2 The Data Area. 7
3.2 The BIRP Bit Map. 8
3.3 The Index File in BIRP. 8
3.4 The Scan Control Table in BIRP. 9
3.5 The BIRP.BRP File 10

4. Main Program - Overview and Overlay Structure. 10

5. Describe 12

6. Search 12

6.1 Parameter Input 13
6.2 Parameter Search. 15
6.2.1 Bitwise Scan 15
6.2.2 Sorted List and Range Scan 16
6.2.3 Random List and Range Scan 17
6.2.4 Area Scan. 18

7. Print. 20

7.1 Add 20
7.2 Clear 21
7.3 Copy. 21
7.4 Delete. 21
7.5 Describe. 22
7.6 Dump. 22
7.7 Micro 22
7.8 Next. 24
7.9 Previous. 24
7.10 Print 25
7.11 Start 25

8. Save 26

9. Restore and Backup 27

10. Examples of BIRP Sessions with Viking Orbiter Library. . . 27

10.1 Log-on and HELP. 27
10.2 DESCRIBE Viking Orbiter Library. 29
10.3 VO Incidence, Emission, Phase Angles similar to Viking Lander Pictures 30
10.4 Box Area Search of Lunae Planum. 31
10.5 Highest Resolution, Clearest Pictures of Mars. . . . 34
10.6 Low Sun Angle Pictures of Tharsis Plains 36
10.7 Limb Shots with Red Filter During Dust Storms. . . . 39
10.8 Seasonal Variations in a Crater Streak in the Elysium Quadrangle 40

11. Autogen. 43

11.1 System Considerations. 43
11.2 Data Structures for Autogen. 44
11.3 Running Autogen. 45
11.4 Program Flow Through Autogen 47

12. Create 49

Appendix 1
Building BIRP on Your System 51

Appendix II
Description of BIRP Routines 54

Appendix III
An Abbreviated Guide to BIRP Commands. 60

1. INTRODUCTION

BIRP (Better Image Retrieval Programs) is a set of programs to interactively sort through and to display a database, such as engineering data for images acquired by spacecraft. This document provides an overview of the philosophy of BIRP design, the structure of BIRP data files, and examples that illustrate the capabilities of the software. The document is divided into twelve main sections. Section 2 is a discussion of the drivers behind the design of the software. Section 3 explains the structure of the BIRP data files. The data file structure is explained before the main processing routines are discussed because BIRP routines interact intensively with these structures. Much of the terminology of BIRP is also defined in section 3. Sections 4 - 9 document each of the main options within BIRP. These sections should give the reader an understanding of the logic and general flow of operation within BIRP. The reader should refer to the source code, which is heavily annotated, if more detail is required. Section 10 illustrates the use of BIRP with examples from the Viking Orbiter library. Sections 11 and 12 discuss how to build a BIRP database. Appendix I contains the information needed: 1) to build and modify BIRP on a DEC RSX-11M system and 2) to guide the systems programmer through the changes necessary to run BIRP on non PDP-11 computers. Appendix II is a list of all the BIRP routines, along with a brief statement about the main function of each routine. Appendix III contains a list of BIRP commands.

2. THE PHILOSOPHY OF BIRP

BIRP was designed to be transportable; that is, relatively easily installed on computers with different operating systems. As such, about 85% of the programs are written in DEC FORTRAN IV+, which is similar to the 1977

ANSI Standard FORTRAN. The remainder, mainly the workhorse routines that are used to do the actual comparisons of data file entries with user defined search criteria, are written in assembly language. The reason for writing these routines in assembly language is to maintain the speed and efficiency needed to examine a large amount of data (example: 50,000 Viking Orbiter data entries) in a time short enough that the system can be properly called "interactive". BIRP is designed to be heavily overlayed so as to run on minicomputers. The task image occupies a total of only 24,000 (16 Bit) words of core. The overlay design allows the main BIRP program, together with several utility routines, to be the only sections of BIRP always in memory. The other routines are called into memory only as needed. BIRP was also designed to be as simple as possible to use. A person with only a limited knowledge of computers should be able to use BIRP after a minimal amount of instruction.

BIRP consists of a main calling program (BIRP.FTN) and about forty subroutines, a few with multiple entry points (Appendix II has listing of routines). Two stand-alone programs, AUTOGEN and CREATE, are used to re-structure a new database so it may be accessed by BIRP. BIRP data files are "parallel" (i.e., there is a separate file for every engineering parameter). As an example, consider the Viking Lander database. There were 1,536 picture taken by the two landers during the primary mission. Twenty-nine parameters for each picture (out of a much larger available set) were extracted and reformatted into BIRP parallel data files. The Nth data entry in each parallel data file contains that parameter value for the Nth lander picture number. When a user searches through a database for all pictures that satisfy a given criterion (such as time acquired, type of filter used, etc.) only one relatively small parameter file need be accessed.

Thus, the user gets "interactive" speed from a minicomputer while accessing large databases.

Before the first search, BIRP creates two bit map files that have one bit for each picture, with the Nth bit corresponding to the Nth picture number. Initially all bits are set to 1 (Figure 2.1). During a search on a particular parameter any picture which does not meet the search criteria has its bit in the bit map set to 0. Subsequent searches on other parameters ignore pictures with their bit map bits set to 0, considerably speeding up the search process. As an example, seaching all 50,000 Viking Orbiter parameter entries takes about 50 seconds on a PDP-11/34 with DEC RL-01 disk drives. Searching 20,000 entries takes about 20 seconds, and searching 1000 entries takes about 2 seconds.

3. BIRP DATA FILE AND BUFFER STRUCTURES

Five types of data files and buffers used by BIRP will be discussed in this section: (1) The PARAMETER.BRP files are the parallel data files mentioned in section 2. They are compared with the user's search criteria during a search procedure. They also provide the engineering data when a user requests a listing of parameter values. (2) The BIRP bit maps (there are two) are scratch files that keep track of which pictures in a library have met the user's search criteria. (3) The index buffer stores the format information about a given parameter file so that parameter can be searched or have its contents output to the user. The data in the index buffer is extracted from the header area of the PARAMETER.BRP file. (4) The scan control table is another buffer within the BIRP programs. It contains some of the index buffer data along with the user's search criteria. The scan control table is accessed by the scanning routines during a search procedure.

(5) The BIRP.BRP file, which is present under that name in every library, contains a description of the library and its parameters.

3.1. THE PARAMETER.BRP FILES

The PARAMETER.BRP files contain the actual database (example: the engineering data for pictures in the Viking Orbiter picture library). The PARAMETER.BRP files consist of a header (which is in addition to any structure created by the operating system or its file control service) and a data area. The header area has information that describes the format of data in the data area, as well as a short text that contains a definition or description of the parameter file. The numeric values of a given parameter are stored in the data area. The parameter value of each picture occupies a separate logical record within the data area.

3.1.1. The Header Area of the PARAMETER.BRP File

The header area of the PARAMETER.BRP file is shown in Table 3.1. Each variable within it will be explained below. The first four entries are the following INTEGER*4 (integers that are four bytes long, hereinafter referred to as I*4) variables:

(1) Length of entire PARAMETER.BRP file in bytes (Bytes 0-3).

(2) Number of pictures (i.e., number of separate records) in the file (Bytes 4-7).

(3) Start byte of description area, counting from beginning of the file (Bytes 8-11).

(4) Start byte of data area, counting from beginning of the file (Bytes 12-15).

These I*4 variables are stored in the PARAMETER.BRP file with the most

significant byte (MSB) first. The DEC standard for integers, however, has the least significant byte (LSB) first. Therefore, the bytes in these I*4 variables must be reversed by the subroutine SCRMBL before they may be used in computations within the BIRP programs. All numeric data in the data area of a PARAMETER.BRP file is stored with the most significant byte first as a standard (this is referred to as BIRP internal format). This format allows fast comparisons of two integers because the MSB bytes of each are compared first. If these bytes are not equal, no further compares need be to made.

(5) Next in the header is the index area, which is used by OPNPRM to form the index buffer. Byte 16 of the file header contains both the width code for the actual length of each record in the data area and the extended information code which identifies certain non-standard data forms. For all types other than bit data, the width of each record in the data area must be a power of 2, so that no record will span a disk block boundary (the DEC RL01 block size is 512 bytes). The legal values of the data width code and the extended information code are listed below.

Data Width Code: Bits 0-2 of Byte 16

Width Code	Actual Data Length
0	1 Bit (thus 8 records/byte)
1	1 Byte
2	2 Bytes
3	4 Bytes
4	8 Bytes
5	16 Bytes
6	32 Bytes
7	64 Bytes

Extended Information Code: Bits 3-7 of Byte 16

Extended Information Code	Extended Information
0	Standard data
1	Data is Latitude
2	Data is Longitude
3-31	Not presently used

(6) Byte 17 contains the I/O conversion type code (also referred to as the data type). The use of the I/O conversion type code allows a wide range of parameter data types to be stored in an efficient manner. For example, most of the real number data (picture center latitude and longitude, etc.) of the Viking Orbiter library are stored as scaled integers, thus cutting the disk storage requirements in half. Another example is text data (Viking Orbiter filter). Text data usually can be stored in one byte by mapping the text into a number with the use of a set of keys. The valid data types are listed below.

I/O Conversion Type Codes: Byte 17

I/O Code	Legal Width Code Range	Description of data
0	0	Single bit ('Y' = 1, 'N' = 2)
1	0	Single bit keyed (use table of keys to decode)
2	1	Single byte keyed (use table of keys)
3	1-4	Integer data (I*1, I*2, I*4, I*8)
4	1-3	Integer data scaled (by R*8 scale factor in the index)
5	1-7	Pure text (1,2,4,8,16,32 or 64 ASCII characters)
6	3-4	Real data (R*4 or R*8, MSB first)

(7) Byte 18 contains the scan code. When searches are made, this code directs BIRP to the proper routine to scan the data. The scan code also controls which type of input (a list of values or ranges) is requested from the user when search criteria are entered. The valid scan codes and the possible I/O codes for each scan code are listed below.

Scan Code: Byte 18

Scan Code	Legal I/O Code	Scan Type
0	0,1	Bit scan
1	1-6	List (data has no numerical order)
2	1-6	Sorted List (data is in numerical order)
3	3-6	Range (data has no numerical order)
4	3-6	Sorted Range (data in numerical order)

(8) Bytes 19-24 contain the FORTRAN output format for the data in the parameter file. For example, the format for the slant range of Viking Orbiter pictures is F8.1. The format is stored as text.

(9) The contents at byte 25 of the header depends on the I/O code. For I/O codes 1 and 2, a set of keys are stored in the header. A key is text that is used (a) to translate the data in the data area into text for output or (b) to translate a text input into a number for a search. For example, the BAND parameter of the Viking Lander library can have two values in the data (0 and 1). These values correspond to "SB" (S-link) and "UH" (UHF-link). The text "SB" and "UH" will be found as keys 0 and 1 in the header of the BAND.BRP file. If keys are present in the header area, byte 25 contains the number of keys. The keys start at byte 26 and the length of each key is stored in byte preceeding the key. If the data type is 4, then bytes 25-32 contan a scale factor as a double precision real number.

(10) At the end of the header area is a description of the parameter, which is stored as text. Byte 36 contains the length of the first description line. There can be up to 32,767 characters (bytes) in the description area.

3.1.2. The Data Area of the PARAMETER.BRP file

The data area of a PARAMETER.BRP data file contains the values of that

parameter for all the pictures in the library. There is one logical record for each picture. The length of each record is determined by the width code, while the format of the data is determined by the data type code. Numerical data are stored with the most significant byte first.

For a database with many pictures, such as the Viking Orbiter data library, choosing the smallest practical data width for each parameter becomes important. The Viking Orbiter mission has about 50,000 pictures. A file like CAMERA.BRP has only two values for a picture, 0 (camera A) and 1 (camera B), and thus uses only one bit per picture, or about 5Kb total. However PICNO.BRP, the picture sequence number, must be text and consequently PICNO entries use 8 bytes per picture or 320Kb total.

3.2. THE BIRP BIT MAP

BIRP utilizes a bit map to keep track of pictures that fulfill the user's search criteria. This is a scratch file (deleted on exit from BIRP) which contains one bit for every picture in the library accessed. There are two bit maps in the BIRP program, one that is modified during a search and one that contains the results of the previous search. The reason for having two bit maps is so that a user can go back one step in a search session. The subroutine INALID sets up the bit maps with all bits set to 1 when the BIRP program is started. Thereafter, during user searches, if the Nth picture in the library fails to meet the search criteria, the Nth bit in the bit map is set to 0. Within a given byte in the bit map the bit order goes from the most to least significant bit.

3.3. THE INDEX BUFFER IN BIRP

The index buffer is used by various BIRP routines to keep track of the

data format of a particular PARAMETER.BRP file. It may have a maximum of 320 bytes if the maximum number of keys (30, each 9 characters long.) is used. The index buffer is diagrammed in Table 3.2. The index is simply the header area of a PARAMETER.BRP file, with some entries expanded to fill whole bytes and with the Integer*4 pointers reversed to DEC standard format (LSB first).

3.4. THE SCAN CONTROL TABLE OF BIRP

The scan control table is a data buffer constructed by REQFLD when a search is to be performed. The scan control table contains information from the index buffer and the user's search criteria. The scan table is passed to the scanning subroutines where it is used during the course of the actual data scan. Table 3.3 is a diagram of the scan control table.

The start of data area, along with the scan code and width code come from the index. Bytes 6-7 contain an Integer*2 number (LSB first) which is the total number of values entered by the user. Since two values are entered for each range in range type scans, this number will be twice the number of ranges. Byte 8 is the width code right-shifted by one, or 2**(byte 8) = width code. Last is the area where the actual values/ranges entered by the user are stored. The values/ranges are stored in the scan control table with the most significant byte first, and the entries are sorted in numerical order. The length of the scan table is 521 bytes. Note that the number of values/ranges that the user can enter depends on the data type. For instance, the scan control table can hold: 512 values (each one byte long), 256 integers (each 2 bytes long), or 64 real number ranges since a range is 2 numbers, and each number is 4 bytes long.

3.5. THE BIRP.BRP FILE

The BIRP.BRP file must be present on every library accessed by BIRP, under that exact name. It is a text file which contains:

LINE NO.

1	Number of pictures (records in file), 17 field type terminated by comma to shorten. May have comments.
2	Name of the library.
3,4	Two explanatory text lines printed out only when entering BIRP.
5...	Any number of lines with descriptions of parameters in library: printed only on main option DESCRIBE command.

There may be a maximum of 80 characters per line. All lines are terminated by carriage return. Table 3.4 is the BIRP.BRP file from the Viking Orbiter library.

4. MAIN PROGRAM - OVERVIEW AND OVERLAY STRUCTURE

The purpose of the main program (BIRP.FTN) is to select, based on the user's request, one of the main processing options. When the BIRP program is started, the first operation performed is to set up the two bit maps by calling the subroutine INALID. At this point all pictures in the current library are made available for searching. Two bit maps are maintained throughtout the BIRP program. One of the bit maps (termed the "active" bit map) is modified when a search is in progress. The other bit map (termed the "inactive" bit map) contains the results of the previous search. When a search is ready to begin, a copy of the active bit map replaces the old inactive bit map. Thus, with two bit maps, the user can backup one search any time while running BIRP (see the BACKUP option).

The main program accepts inputs from the user through the subroutine REQUST. Since REQUST is used to interpret all of the user's inputs, it will

be discussed in some detail. Its purpose is to print a prompt for the user, which is provided by the calling routine, and to return the user's input as a string of text. In addition, REQUST checks the input for some special answers and returns a flag if one is detected. The special answers REQUST checks for are the following: (1) DONE: the user is finished with the present section; (2) ABORT: the user has made a mistake in the present section; (3) EXIT, BYE or OFF: the user wants to exit from the BIRP program; (4) HELP: the user requests that a message be printed that explains the possible inputs and (5) the user entered a blank line. The action taken when the user enters one of the special answers is determined by the calling routine.

When a valid command is input, the main program will direct the operation of the program to the requested option. After that operation is completed, control is transferred back to the command request section. The main processing options available to the user are: DESCRIBE, SEARCH, PRINT, SAVE, RESTORE and BACKUP. Each option will be discussed in detail in the following sections.

The BIRP task is overlayed so that it can fit into the small memory size of a minicomputer. The overlay structure is like a tree structure, in that there is a base or root segment which is always in the computer memory, and a series of branches which are read into the memory from a disk as they are needed. When a branch is brought into the memory it will overlay or replace the previous branch that was resident in memory. The base segment of BIRP consists of the main program and all the subroutines that are used frequently throughout the program. Each of the main options, except for restore and backup, are separate branches in the overlay structure. In addition, some of these branches have sub-branches (Table 4.1). As such, the memory

requirement for BIRP is determined by the size of the root and the size of the largest branch.

5. DESCRIBE

The DESCRIBE option will print out the name of the library currently being accessed, the number of pictures in the library and a list of the searchable parameters included in the library. A brief description of each parameter is also provided for the user. The describe option calls the subroutine OPNBRP, which prints out the contents of the BIRP.BRP file. If the BIRP.BRP file is missing or incorrectly formatted, then an error message is printed on the user's terminal and OPNBRP exits to the operating system. Otherwise, after the information from the BIRP.BRP file is listed, OPNBRP returns to the main program, where the next main processing option will be requested.

6. SEARCH

The SEARCH option of BIRP performs searches of parameter files to find pictures that meet user defined criteria. It can be divided into two subsections. The first subsection sets up a search by requesting the name of a parameter file that is to be searched along with a list of values or a set of ranges to search for within that parameter file. The second subsection does the actual searching of the parameter files. There are six possible methods of scanning the parameter files: bitwise, sorted lists and ranges, random lists and ranges, and area searches. The method of scanning used for a particular parameter file depends on the data type (see section 3.1) of that file. If a search is attempted when there are no pictures available or if a search finds no pictures that meet the search criteria, then control is

returned to the main program. When a parameter name is requested, DONE must be entered to signal that the user is finished searching the database. Control is then returned to the main program, where a new main processing option will be requested.

6.1. PARAMETER INPUT

The first information requested in the search section is a parameter name. After a parameter name is entered, the subroutine OPNPRM is called to open the parameter file and perform some simple checks. If the selected parameter file does not exist, an error message from OPNPRM is printed on the user's terminal. The user is then asked for a new parameter name.

OPNPRM reads the index (see section 3.3 for details on the index) from the file header. OPNPRM checks for the following items: (a) The parameter file is in the proper order. (b) The index is the correct size. (c) The values for the data length, data type and scan code are valid and internally consistent (e.g. if the data type is 2, then OPNPRM checks that the data width code is 1). and (d) The number of keys, if keys are present, is correct and all the keys have a valid length. The current file is closed and a new parameter name is requested if any errors are found in the index.

Either the subroutine REQFLD or AREFLD is called after the parameter file has been opened and checked. The purpose of these two routines is to request a set of values or ranges from the user that define the search criteria. The scan control table (see section 3.4) is then set up, based on the inputs from the user. The type of input (text, values or ranges) depends on the data type and scan code for the parameter file being accessed. AREFLD is called if the scan code is 5 or 6 (area type searches). REQFLD is called for any other value of the scan code. Table 6.1 shows the type of input

requested for each of the possible data scan codes.

Inputs recieved by REQFLD are converted to the proper internal BIRP format by using the subroutine CVTIN. For example, if the data type is 2 (text that is translated by a key into a number), then the text input by the user is compared with the list of keys. If a match is found, the input text is converted to the number that corresponds to the proper key. If a match is not found a BIRP input conversion error is reported and a new value is requested. REQFLD also sorts the entries in the scan control table from the smallest to the largest value. This is done to increase the speed of the searches. Speed is increased, since on the average a given picture does not have to be compared with all the entries in the scan control table. For example, if the parameter value of a given picture is less than the first entry in the scan control table, then the parameter value will be less than all the entries in the scan control table because the entries are sorted in numerical order. Thus, only one comparsion needs to be performed for that picture. REQFLD checks that values are not entered more than once or that a new range does not overlap with a previous range. An error is reported and the input is ignored if one of these conditions is detected. The user can enter any number of values or ranges for a particular search until the scan control table is full. However, the scan control table is large enough so that it is rarely, if ever, full. The user enters DONE when a value or range is requested to inform the program that there are no more inputs for that search. When DONE is recieved by REQFLD, control is transferred to the proper search routine.

Parameter value inputs for area type searches are handled by a separate subroutine (AREFLD) because they are special cases. The number of values that are input is fixed. For a point search, a single latitude and longitude

pair is requested. For a box or area search, exactly four pairs of latitudes and longitudes must be input to define the search area. AREFLD does not sort the inputs. The searching starts immediately after the four pairs of latitudes and longitudes for an area search or the one pair for a point search are input. DONE can not be input to start an area search. AREFLD will ignore any DONE that is input. The scan control table is set up by AREFLD just as REQFLD does, except for what has been mentioned above. The latitudes and longitudes pairs are placed in the scan control table where the parameter values or ranges would be placed by REQFLD.

6.2. PARAMETER SEARCH

Each of the six modes of BIRP searches, bitwise, sorted lists and ranges, random lists and ranges, and area searches are discussed below. Since the search procedure for lists and ranges are similar, the sorted list and sorted range methods will be discussed together, as will the random list and random range methods.

6.2.1. Bitwise Scan

The bitwise method of searching is used for bit type data files (scan code of 0) and the searching is done with the subroutine BITSCN. A bit type data file is one where there are only two possible values of the parameter. Viking Orbiter A or B is an example of bit data. A bit data file is created when the SAVE option (section 8) is used, where the possible values are that a given picture did or did not fulfill the search criteria. Each bit in the parameter file represents one picture in the library. The bitwise scan is the fastest type of search because the pictures are searched eight at a time and the search involves simple comparisons.

The search procedure starts as soon as the user inputs one of the two possible values for the parameter (Viking Orbiter A or B; or yes/no), without the user having to enter DONE. The input from the user is converted to a byte where all eight bits are either 0 or 1. The search, which is a two step procedure, is illustrated in Table 6.2. The first step is a logical "exclusive or" comparison between the parameter value for a picture and the user input (which is now coded as a 0 or 1). The "exclusive or" function returns a value of 1 if, and only if, the parameter value and the user's input are not the same. Thus, in order for the "exclusive or" step to work, the user's input is coded into the opposite value of what is being search for in the parameter file. For example, if the parameter value being searched for is a 1, then the user's input will be coded as a 0 and vice versa. The second step is a logical "and" between the result of the first step and the current bit map. BITSCN does the logical comparisons by using the DEC FORTRAN IV+ functions IEOR and IAND.

6.2.1. Sorted List and Range Scan

The sorted search method is used for parameter files where the data in the parameter file is sorted from the smallest to the largest value. The term list implies that the number of possible values of the parameter is limited and the user is searching for a subset of the possible values. The term range indicates that the data are taken from a continuous distribution of numbers and that the user is looking for all pictures with values between a pair of specified values. List will be used throughout the following discussion but range can be substituted for it. The sorted list search is performed by the subroutine LSTSEQ and the sorted range by RNGSEQ. There are no examples of these type of data files in the Viking Lander or Orbiter data

sets. If each Viking Orbiter was a separate data set, then the revolution number would be an example of a sorted range data file.

Only those pictures still selected (not yet eliminated by a previous search) are tested. The data file is searched from the beginning until a picture is found with the same value as the first (smallest) entry in the scan control table. Any picture in the data file with values less than the first entry in the scan control table have their bits in the bit map set to 0 (Table 6.3). These pictures can not match any other entry in the scan control table because it is sorted in numerical order. The bits in the bit map that correspond to pictures with a value that matches the first entry remain as a 1. When the first picture with a value greater than the first entry in the scan control table is found, the next entry is fetched from the scan control table. The above procedure continues until the first picture with a value greater than the last (largest) entry in the scan control table is found. The rest of the pictures in the parameter file will have their bits in the bit map set to 0, since they are greater than all the entries in the scan control table.

6.2.3. Random List and Range Scan

The random search method is used for data files where the values within the file are not in any particular order. Refer to the previous section for an explanation of the terms list and range. The searching is performed with the subroutines LSTRND and RNGRND. Viking Orbiter filter number is an example of a random list data file. Solar incidence angle is an example of a random range data file.

Again only pictures which have not been eliminated by a previous search are tested. The value of each picture is compared with each entry in the

scan control table. There are three possible outcomes for the comparisons. First, if the value for the picture is greater than the current entry from the scan control table, then the next entry in the scan control table is tested. Second, if the picture is equal to the current entry, then that picture remains selected and the next picture in the data file is tested. Third, if the picture is less than the current entry, then there can not be a match because any entry left in the scan control table is greater than the current entry. When the third outcome occurs the bit in the bit map that corresponds to the current picture is set to 0 and the test for the next picture in the data file is started.

6.2.4. Area Scan

With the exception of the area search, all BIRP searches are done by the use of comparisons, either between a range or a set of ranges, or between a list (value or values) specified by the user, and a given parameter value. Area search is based on the picture corner coordinates (latiude and longitude or azimuth and elevation). These data are used to test whether: (1) a given picture contains a user specified location (point search) or (2) a given picture shares any common area with a user specified 4-sided polygon (area search). The area type search is performed by the subroutine ARESCN.

Before beginning the time consuming spherical trigonometric computations needed to tell whether a picture lies within a given area, ARESCN tries to eliminate the picture by comparing the maximum and minimum latitudes and longitudes of the picture coordinates with the user specified area. If a given picture is entirely outside the box defined by the maximum and minimum latitudes and the maximum and minimum longitudes of the search area, then that picture is removed from the list of searchable pictures. Thus, the

lengthy spherical trigonometric computations are avoided for many of the pictures. If the picture is within the previously described box, then the procedure discussed below is used to determine if the picture is within the actual user specified area. Note that this preprocessing is only done if the user has specified a box or polygon.

To test whether a given point is within a polygon, four angles are contructed using spherical trigonometry. Each angle is formed by using the search point and adjacent points of the polygon with the search point as the apex. If the point is within the polygon, then the sum of the four angles will be 360 degrees. If the point is outside the polygon, then the 4 angles will have a sum of 0 degrees. When a point search is performed, the user defined latitude/longitude point is tested by the above procedure to see whether the point is within the area defined by the corner coordinates of each picture.

In the area search mode, the problem is to determine if there is any overlap between two polygons, one being a given picture and the other being the user specified search area. Up to eight points are checked with the above procedure. Each corner of the picture is checked to see if it is within the search area. Then each corner of the search area is checked to see if it is within the picture. When one corner is found to be within the appropriate polygon, the picture is retained in the list of searchable pictures, and the next picture is tested. It is possible to have two areas overlap and be rejected by the area search algorithm. This arises when none of the corner points of the picture are within the search area and none of the corner points of the search area are within the picture. Testing for this situation would greatly increase the area search processing time. Since ARESCN is already slow, this small "bug" was not fixed. Furthermore, this

situation should not occur if the picture and the search area are not equal in size.

7. PRINT

PRTBRP is the subroutine within BIRP that controls the printing of data and viewing of microfiche frames. It is called by the main program BIRP when the user enters the PRINT main processing option. There are 11 main commands in this subroutine: ADD, CLEAR, COPY, DELETE, DESCRIBE, DUMP, MICRO, NEXT, PREVIOUS, PRINT and START. Each command is discussed in separate sections below. These commands use two main data structures: the print list and the picture list. The print list is a sequential scratch file which has a list of the parameter names to include when data is printed (example: CLAT, CLONG, FILTER for Viking Orbiter library), in text form, one name per record. The picture list is the present bit map resulting from a series of searches by the user. Figure 7.1 is a schematic illustration of commands within the PRINT section of BIRP.

7.1. ADD

ADD is a command in PRTBRP which places a parameter name in the print list. ADD requires a parameter as an argument, separated from ADD by a space. In PRTBRP, FNDSPE is used to separate the parameter name argument from the ADD command on the user's input line. PRTBRP calls OPNPRM using this parameter name to test whether that particular PARAMETER.BRP file is present on the user's account. If the file is present and uncorrupted, PRTBRP writes the parameter name in text form to the print list. Thus when the user chooses to print data for selected pictures, the data for the parameters added will appear on the output.

7.2. CLEAR

CLEAR is a command in PRTBRP which deletes all entries on the print list. CLEAR simply closes and deletes the print list file, and reopens it again as a new scratch file. The picture number parameter (exact name: PICNO.BRP) always remains as the default (first entry) on the print list.

7.3. COPY

COPY is a command in PRTBRP which places parameter names from a file (previously created by the user outside the BIRP program) into the print list. This text file contains the parameter names the user wishes data printed for. Copy requires a valid RSX-11M file name as an argument. This file must contain legal parameter names separated by carriage returns. The parameter names in this file are then written into the scratch file containing the print list. The parameter names copied are in addition to any names added to the print list by the user. These copied names are not checked for validity as are parameter names inserted into the print list by the ADD commend.

7.4. DELETE

DELETE is a command in PRTBRP which removes the picture presently accessed in the picture list (bit map). As an example, if the user prints data for a picture and finds it necessary for some reason to eliminate the picture from the picture list, DELETE would be used. DELETE calls CLRPID which sets the bit in the bit map of the presently accessed picture to 0. Then, PRTBRP gets the picture ID of the next selected picture.

7.5. DESCRIBE

DESCRIBE is a command in PRTBRP (and in SEARCH) which prints the description area text from a PARAMETER.BRP file. The user types DESCRIBE and the parameter name separated by a space. PRTBRP uses FNDSPE to separate the argument and then uses that argument in a call to OPNPRM to print the description area text. If there is no PARAMETER.BRP file of that name or the file is corrupted, an error message is printed.

7.6. DUMP

DUMP is a command in PRTBRP which prints data for the parameters in the print list for all selected pictures (the current picture list). DUMP first directs that a header with the parameter names be printed. Then it sets the picture ID number so the next picture accessed will be the first selected picture. Each PARAMETER.BRP file whose name is in the print list is successively opened, the data converted to text by reference to a key or real scale factor, if necessary, and the converted value written to an array. The parameter values for each picture are printed on one or more lines, with each containing a maximum of ten parameters. When all parameters in the print list have been used, PRTBRP calls GNXPID again to find the picture ID of the next selected picture. This continues until the end of the picture list is reached. When this happens GNXPID returns a picture ID value of -1. The DUMP is then complete and PRTBRP requests a new input command from the user.

7.7. MICRO

MICRO is a command in PRTBRP which permits the user to view microfiche frames on an automatic microfiche viewer such as the Image Systems Card Viewer. The MICRO command takes an argument of ON or OFF which enables or

disables, respectively, viewing of microfiche frames. Enable sets a flag (the variable MIC) to 2. Disable sets the flag to 0 and closes a file that contains the microfiche machine's I/O number. After setting the MIC flag, MICGO is called. This subroutine uses REQUST to ask the user for the physical device name of the microfiche reader (example TT1:). TT2: is the default device for the microfiche reader. MICGO opens a file with this physical device name as the file name.

If the file is opened successfully, MICGO calls FCHEDT, the subroutine which allows the user to manipulate microfiche frames. FCHEDT calls VIEW on the first entry and VIEW checks the fiche code to see if the fiche previously selected (if any) by the user is the same as the presently requested fiche frame. If the frame that is requested is the same as the one currently displayed, BINFLC an entry point in the routine FCHLIB is called to translate the row/column coordinate of the frame requested to the command code required by the microfiche machine. This code is returned to VIEW, which sends it to the microfiche viewer. VIEW then returns to FCHEDT. If a new microfiche card needs to be displayed, VIEW calls BINFCH first to convert the micofiche fiche code to the command form, then calls BINFLC to interpret the row/column data before sending the data to the microfiche machine.

After returning to FCHEDT, the user may use any of the standard PRINT commands to display a microfiche frame for any picture from the picture list). FCHEDT has six other options for manipulating microfiche cards. The user may display other positions on the presently selected microfiche card by moving the card up (^), down (-), right (>), or left (<) one row or column at a time using the four symbolic commads in parenthesis. The actual microfiche coordinate (example: H5) or microfiche card number (example: 6S4) may be entered to view that particular frame or card. These six commands pass a

modified value for the row/column number and/or microfiche card number to VIEW which uses BINFCH and BINFLC to interpret the inputs.

When the user is finished viewing microfiche frames and wishes (for example) to print a list of data values for selected pictures, the microfiche viewing may be stopped by the command MICRO OFF. This action avoids the time delay caused by the slow speed of the microfiche reader's search cycle, and allows another user access to the reader.

7.8. NEXT

NEXT is a command in PRTBRP that increments the picture ID number and prints the data for the next picture in the picture list. When NEXT is input, PRTBRP calls GNXPID to find the picture ID of the next selected picture. The PARAMETER.BRP files whose names are in the print list are opened by OPNPRM. The data for each picture are then converted to text by reference to a key or real scale factor, if necessary, and PRTBRP writes these converted values to an array. Note that when ten parameter values have been read into the array (including PICNO.BRP which always comes first on the first line printed) the line of values is printed before more parameter names from the print list are accessed.

7.9. PREVIOUS

PREVIOUS is a command in PRTBRP which decrements the picture ID number and prints the data for the picture before that presently selected in the picture list. When PREVIOUS is input, PRTBRP calls SETSCN to set a pointer to the beginning of the picture list. The picture list is scanned by calling GNXPID the requisite number of times to access the proper picture. If the picture ID was set to the first picture in the picture list, PREVIOUS will

"wrap around" so the next picture accessed is the last picture in the picture list. PRTBRP then prints the data values for the parameter names in the print list as with a NEXT command.

7.10. PRINT

PRINT is a command in PRTBRP that prints data values for any number of pictures in the picture list going either up or down the list. PRINT has the argument of +N or -N, where N is the number of pictures to print. The default is N = 1, which prints the next picture on the picture list. PRTBRP separates the argument from the main command PRINT by calling FNDSPE and assigns this signed number to the variable NUMPRT (default is +). As with PREVIOUS and NEXT, PRTBRP calls the proper routines, GNXPID or SETSCN AND GNXPID after decrementing or incrementing the picture ID. As each picture ID is accessed the data values for the parameter names in the print list are printed, and NUMPRT is decremented by 1 if > 0 or incremented by 1 if < 0. Succeeding picture's data values are printed and if the start of the list is reached a note is printed and the next picture accessed is the last in the picture list. As with NEXT if the last picture in the list is reached, the next picture accessed will be the first picture in the picture list.

7.11. START

START is a command in PRTBRP which sets the picture ID to the beginning of the picture list. PRTBRP calls SETSCN to set pointers to the start of the list, and then sets the picture ID to -1. GNXPID is called so that the next picture accessed is the first picture in the picture list. Its parameter data values are printed as with NEXT.

8. SAVE

The main processing option SAVE is used to create a BIRP parameter data file out of the current bit map. The newly created parameter file is a bit type data file. This option is useful for quickly reproducing long and complicated searches. It can also be used to create a parameter file of pictures containing a certain feature (see section 10.8 for an example of the SAVE option). The SAVE option utilizes the subroutine SAVBRP.

SAVBRP initially requests a name for the data file that is to be created. The new name is checked to make sure it is not already used for a data file. If the name is being used, SAVBRP requests a new file name. Once the file name is accepted, SAVBRP will create two of the three files used by CREATE to make a BIRP data file (see section 12 for details about the files needed by CREATE). The bit map is saved as a PARAMETER.DAT file. A PARAMETER.IDX file, which contains the information necessary to create the header area of the PARAMETER.BRP file, is also built by SAVBRP. Specifically, the PARAMETER.IDX file contains the following data:

The number of pictures in the library.

Data wide code of 0 (1 bit per picture).

No extended information.

Data type code of 0 (bit data mapped by 0=no and 1=yes).

Data scan code of 0 (bitwise scan).

Output conversion format of 1A1.

SAVBRP then informs the user to edit a text file (the PARAMETER.TXT file used by CREATE) that describes the new parameter file. Finally, SAVBRP closes all open files and exits to the operating system. If the user enters DONE or ABORT when the parameter name is requested, SAVBRP will return control to the main program without creating any files.

9. RESTORE AND BACKUP

RESTORE is a main processing option that allows the user to make all of the pictures in the current library available for searching. BACKUP is a main processing option that allows the user to recover when a mistake has been make during a search or when the user simply does not like the results of a particular search. Both of these options are performed within the main program. Control of the program is returned to the main processing option request section after the completion of both of these options.

The RESTORE option simply calls the subroutine INALID. INALID deletes the two current bit maps and creates two new bit maps with all the bits in the map set to 1. The BACKUP option replaces the active bit map with the inactive bit map. The active bit map contains the results of the most recent search and the inactive bit map contains the results of the second most recent search. The number of selected pictures in the new active bit map is counted and reported to the user. Note that the user can not go back more than one step because only two bit maps are used.

10. EXAMPLES OF BIRP SESSIONS WITH VIKING ORBITER LIBRARY

Following are 8 examples illustrating BIRP commands for the Viking Orbiter Library. User entries are shown in lower case. Comments designed to elucidate the philosophy behind the search sequences are tabulated before the beginning of the actual sessions.

10.1. Log-on and HELP:

The example is designed to show user how to log-on to RSX-11M operating system. Note that all user entries must be followed by carriage return. Log-on consists of entering the account, which is ORBIT and the password which is BIRP. For security

BIRP is not echoed on the terminal. HELP command in this case provides general message describing main processing options and universal options.

```
>hello
ACCOUNT OR NAME: orbit
PASSWORD:

        RSX-11M BL22    MULTI-USER SYSTEM

GOOD MORNING
20-NOV-79 11:13 LOGGED ON TERMINAL TT0:

WELCOME TO THE IMAGE PROCESSING SYSTEM AT THE
WASHINGTON UNIVERSITY REGIONAL PLANETARY IMAGE FACILITY
>@LOGIN.CMD
>ASN DL1:=BP1:
>;TO START "BIRP" TYPE: RUN BIRP
>@ <EOF>
>run birp


BETTER IMAGE RETRIEVAL PROGRAMS

YOU ARE ACCESSING THE VIKING ORBITERS 1 AND 2 LIBRARY
IN WHICH THERE ARE     39757 PICTURES
FRAMES 003A01 TO 974A10 AND 004B01 TO 705B52 ARE INCLUDED IN THIS LIBRARY.
TYPE "HELP" IF YOU'RE NEW TO BIRP.



ENTER MAIN PROCESSING OPTION:  help


YOU ARE RUNNING THE BETTER IMAGE RETRIEVAL PROGRAM
     THIS SECTION HAS THE FOLLOWING OPTIONS:
SEARCH      -START UP A SEARCH OF PARAMETER FILES
PRINT       -PRINT RESULTS OF A SEARCH
SAVE        -SAVE THE PRESENT PICTURE LIST AS A BIRP FILE
RESTORE     -RESTORE PICTURE LIST TO 'ALL SELECTED'
DESCRIBE    -PRINT OUT DESCRIPTION OF PARAMETER FILES
BACKUP      -GO BACK TO PICTURE LIST FROM PREVIOUS SEARCH
    NOTE:   YOU NEED TYPE ONLY THE FIRST 3 LETTERS OF ABOVE COMMANDS

    IN RESPONSE TO ANY QUERY IN THIS SYSTEM YOU MAY TYPE:
DONE        -DONE WITH PRESENT PART, BEGIN PROCESSING
ABORT       -IF BAD RESPONSE IS MADE, RETURN TO LAST OPTION
HELP        -DESCRIBE WHAT TO DO
EXIT, BYE, OFF  -LEAVE  BIRP  SYSTEM
```

10.2. Describe:

This example is designed to show to the user BIRP's response to the DESCRIBE option for the Viking Orbiter Library. After the DESCRIBE is finished printing, BIRP automatically requests a Main Processing Option. BOXNO consists of the 10 by 10 degree box number that a given picture center latitude and longitude falls within. The number sequence begins at 80-90 degrees N. lat., 0 to 10 degrees W. long. as BOXNO 1. The numbering sequence proceeds clockwise in longitude when viewed from the north and then southward in latitude. The box located at 80-90 degrees S. latitude, 350-360 degrees W. longitude is BOXNO 648.

```
ENTER MAIN PROCESSING OPTION:  describe

BETTER IMAGE RETRIEVAL PROGRAMS

YOU ARE ACCESSING THE VIKING ORBITERS 1 AND 2 LIBRARY
IN WHICH THERE ARE   39757 PICTURES

PARAMETERS
----------
BOXNO   10 DEGREE BOX NUMBER CONTAINING CENTER OF PICTURE.    [RANGE: 1 - 648]
CAMERA  CAMERA USED: WHICH CAMERA OF ORBITER TOOK PICTURE.    [VALUES: A OR B]
CLAT    LATITUDE OF PICTURE CENTER: +90=N.POLE,-90=S.POLE [RANGE:-90.0 TO +90.0]
CLONG   WEST LONGITUDE OF PICTURE CENTER:                     [RANGE:0-360.0]
CRAFT         SPACECRAFT: WHICH VIKING ORBITER.                       [VALUES:1 OR 2]
EMANG   EMISSION ANGLE: NORMAL TO SURFACE = 0 DEGREES         [RANGE:0-90.0]
FILTER  FILTER USED [VALUES: BLUE, MINUS (-BLUE), VIOLET, CLEAR, GREEN,OR RED]
INANG   INCIDENCE ANGLE: NORMAL TO SURFACE = 0 DEGREES                [RANGE:0-90.0]
LATLON  AREA SEARCH FILE: NON PRINTABLE
LS      AEROCENTRIC LONGITUDE OF THE SUN: POSITION OF SUN     [RANGE:0-360.0]
MCNO    MARS CHART NUMBER:MARS MAPPING CHART (USGS QUADRANGLE) [VALUES:1 TO 30]
PHANG   PHASE ANGLE:ANGLE BETWEEN EMISSION AND INCIDENCE ANGLES       [RANGE:0-180.0]
PICHT   PICTURE HEIGHT: HEIGHT OF PICTURE FRAME IN KILOMETERS.        [RANGE:2-2000]
PICNO   PICTURE SEQUENCE  : REVOLUTION, SPACECRAFT, PICTURE COUNT IN ORBIT.
        STANDARD PICTURE SEQUENCE NUMBER: XXXYZZ, WHERE XXX=ORBIT OR REVOLUTION
        NUMBER, Y=SPACECRAFT(A=VIKING 1, B=VIKING 2, S=VIKING 1 SURVEY MISSION)
        ZZ=PICT COUNT WITHIN ORBIT
PICWD   PICTURE WIDTH: WIDTH OF FRAME IN KILOMETERS.          [RANGE:2-2000]
QUAL    PICT. QUALITY (DEGREE OF HAZINESS) [VALUES:CLEAR,SLIGHT,MODERATE,DENSE]
RES     RESOLUTION: SIZE OF A PICTURE ELEMENT (PIXEL) IN METERS.[RANGE:1-1500]
REVNO   REVOLUTION NUMBER: ORBIT NUMBER OF SPACECRAFT         [RANGE:0-999]
SLANTR  SLANT RANGE: VO TO PICT. CENTER ON MARS/SATELLITE (KM).[RANGE:250-30000]
TARGET  SUBJECT. [VALUES: PHOBOS, DEIMOS, MARS, STAR, TERM(TERMINATOR) OR LIMB]
```

10.3. Viking Orbiter Frames With Incidence, Emission, Phase Angles Similar to Viking Lander Frame:

The center of Viking Lander color frame 12A048 has a solar incidence angle of 7 degrees, an emission angle of 57 degrees, and a phase angle of 52 degrees. The following search of Viking Orbiter frames was designed to find Orbiter frames with incidence, emission and phase angles within 5 degrees of the values for the Lander frame center. Use of DESCRIBE <PARAMETER>, additional HELP messages, and construction of a print list is illustrated. The intent of such a search could be, for instance, to find VO frames that can be directly compared to the Lander frame in terms of colorimetry.

```
ENTER MAIN PROCESSING OPTION: search

THERE ARE   39757 PICTURES PRESENTLY SELECTED

PARAMETER: help


PLEASE ENTER THE NAME OF A PARAMETER YOU WISH TO SCAN.
"DESCRIBE (PARAMETER)" FOR A DESCRIPTION. EXAMPLE:  "DESCRIBE TARGET"
NOTE: TYPE "DONE" THEN "DESCRIBE" FOR A LIST OF PARAMETERS

PARAMETER:  describe target
TARGET SUBJECT OF PICTURE: PHOBOS, DEIMOS, MARS OR STAR (OFF PLANET)
PARAMETER:  target
VALUE   1:  help


YOU MAY NOW SELECT VALUES TO SEARCH FOR.

PROGRAM PROMPT                  INPUT REQUIRED
--------------                  --------------
YES (Y) OR NO (N)               "Y": PARAMETER PRESENT (TRUE),"N": ABSENT (FALSE)
WHICH ONE ?                     [TEXT] ONE OF TWO POSSIBLE VALUES.
VALUE (N):                      [VALUE] TEXT OR NUMBER, DEPENDING ON PARAMETER.
RANGE (N) LOW LIMIT:            [VALUE]: A NUMBER, THE LOWER RANGE LIMIT.
RANGE (N) HIGH LIMIT:           [VALUE]: A NUMBER, THE UPPER RANGE LIMIT.


NOTE: 1) AFTER LOWER LIMIT, "DONE" FOR UPPER LIMIT SEARCHES TO MAX.
      2) "(N)" IS THE NUMBER OF VALUES OR RANGES YOU HAVE ENTERED.

VALUE   1:  mars
VALUE   2:  done
```

```
THERE ARE    35655 PICTURES PRESENTLY SELECTED

PARAMETER:  inang
RANGE   1 LOW LIMIT:  2
RANGE   1 HIGH LIMIT: 12
RANGE   2 LOW LIMIT: done

THERE ARE     282 PICTURES PRESENTLY SELECTED

PARAMETER:  emang
RANGE   1 LOW LIMIT:  52
RANGE   1 HIGH LIMIT:  62
RANGE   2 LOW LIMIT:  done

THERE ARE     17 PICTURES PRESENTLY SELECTED

PARAMETER:  phang
RANGE   1 LOW LIMIT:  47
RANGE   1 HIGH LIMIT:  57
RANGE   2 LOW LIMIT:  done

THERE ARE      9 PICTURES PRESENTLY SELECTED

PARAMETER:  done
ENTER MAIN PROCESSING OPTION:  print
ENTER PRINT REQUEST:  add clat
ENTER PRINT REQUEST:  add clong
ENTER PRINT REQUEST:  add inang
ENTER PRINT REQUEST:  add emang
ENTER PRINT REQUEST:  add slantr
ENTER PRINT REQUEST:  print 9
```

PICNO	CLAT	CLONG	INANG	EMANG	PHANG	SLANTR
669A16	13.92	302.76	10.30	52.88	53.66	34330.0
723A81	24.82	226.74	8.62	60.06	50.82	22354.0
241B44	-28.16	312.48	9.28	55.62	56.62	23890.0
248B44	-23.42	112.26	6.74	58.26	54.62	24188.0
248B70	-19.06	100.72	6.74	59.20	55.30	24178.0
254B25	-16.68	284.56	9.10	57.76	47.78	26934.0
254B52	-18.46	271.64	9.88	52.92	47.94	26706.0
292B26	-13.24	278.14	11.94	60.72	47.56	27684.0
308B25	-22.30	192.80	4.98	53.02	51.20	27300.0

```
ENTER PRINT REQUEST :done
```

10.4. Box Area Search of Lunae Planum:

This search is meant to illustrate the use of the box mode of the area search. The RESTORE command is used to set all the bits in the bit map to 1 (i.e. all

pictures are selected). The user then restricts the active picture list to Mars pictures located on MCNO 18. (MCNO 1 extends from the north pole to 65 degrees N. latitude. MCNO's 2 to 7 extend from 30 to 65 degrees N. latitude, and they each span 60 degrees in longitude. MCNO's 8 to 15 extend from 0 to 30 degrees N. latitude, and they each span 45 degrees in longitude. MCNO's 16 to 30 are the equivalent areas in the southern hemisphere). LATLON, which gains entry into the area search, is then entered, and HELP is used to explain the options. ABORT is used to halt the input of box corner points and LATLON is called again. Pictures that fulfill the area search constraints share a common area with the user defined box. The RES and QUAL parameters are then used to further restrict the picture list. DUMP is used to illustrate a default printing of the picture list, which contains only picture numbers (PICNO). The final picture list consists of those frames over a highly fractured area in Lunae Planum.

```
ENTER MAIN PROCESSING OPTION: restore
ENTER MAIN PROCESSING OPTION: sea

THERE ARE   39757 PICTURES PRESENTLY SELECTED

PARAMETER: target
VALUE   1: mars
VALUE   2: done

THERE ARE   35655 PICTURES PRESENTLY SELECTED

PARAMETER: mcno
VALUE   1: 18
VALUE   2: done

THERE ARE    1566 PICTURES PRESENTLY SELECTED

PARAMETER: latlon
ENTER AREA SEARCH MODE:help
YOU ARE IN THE AREA SEARCH SECTION OF THE
BETTER IMAGE RETRIEVAL PROGRAM.  YOUR OPTIONS ARE:
   POINT -      YOU WISH TO SEARCH FOR A GIVEN POINT
   AREA  -      YOU WISH TO SEARCH FOR A SPECIFIED BOX
ENTER AREA SEARCH MODE:area
LONGITUDE (AZIMUTH) OF POINT 1: 69
LATITUDE (ELEVATION) OF POINT 1: 25
LONGITUDE (AZIMUTH) OF POINT 2: 75
```

```
LATITUDE (ELEVATION) OF POINT 2: abort
PARAMETER: latlon
ENTER AREA SEARCH MODE: area
LONGITUDE (AZIMUTH) OF POINT 1: 69
LATITUDE (ELEVATION) OF POINT 1: -25
LONGITUDE (AZIMUTH) OF POINT 2: 75
LATITUDE (ELEVATION) OF POINT 2: -28
LONGITUDE (AZIMUTH) OF POINT 3: 78
LATITUDE (ELEVATION) OF POINT 3: -15
LONGITUDE (AZIMUTH) OF POINT 4: 67
LATITUDE (ELEVATION) OF POINT 4: -14

THERE ARE     223 PICTURES PRESENTLY SELECTED

PARAMETER: res
RANGE   1 LOW LIMIT: 0
RANGE   1 HIGH LIMIT: 230
RANGE   2 LOW LIMIT: done

THERE ARE      54 PICTURES PRESENTLY SELECTED

PARAMETER: qual
VALUE   1: clear
VALUE   2: slight
VALUE   3: moderate
VALUE   4: done

THERE ARE      43 PICTURES PRESENTLY SELECTED

PARAMETER: qual
VALUE   1: clear
VALUE   2: done

THERE          34 PICTURES PRESENTLY SELECTED

PARAMETER: done
ENTER MAIN PROCESSING OPTION: print
ENTER PRINT REQUEST: dump

063A21
063A24
464A12
464A13
464A14
464A15
464A16
464A17
464A18
464A19
464A20
464A21
464A22
464A23
608A25
608A26
```

```
608A27
608A28
608A29
608A30
608A32
608A46
608A47
608A48
608A49
608A50
608A51
608A52
608A53
608A69
608A71
610A01
610A02
610A03

END OF LIST REACHED: RESETTING TO START
ENTER PRINT REQUEST: done
```

10.5. Highest Resolution, Clearest Pictures of Mars:

This search is designed to find the pictures of Mars taken at the highest resolution (0 to 10 meters/pixel), under good seeing conditions. The picture list was further constrained by use of the FILTER parameter, which showed that neither the red nor the green filters were used at high resolution. The search also illustrates the use of an Image Systems microfiche reader with BIRP. Use of the NEXT or PREVIOUS commands are shown to go to the next or previous picture in the picture list, respectively. Use of the PRINT <N> command shows the Nth picture from the present picture being displayed. Use of the DELETE command is illustrated for removing a picture from the picture list. This could have been done, for example, in response to examining the microfiche images to further restrict the search.

```
ENTER MAIN PROCESSING OPTION: restore
ENTER MAIN PROCESSING OPTION: search

THERE ARE   39757 PICTURES PRESENTLY SELECTED

PARAMETER: target
VALUE    1: mars
VALUE    2: done
```

```
THERE ARE   35655 PICTURES PRESENTLY SELECTED

PARAMETER: res
RANGE   1 LOW LIMIT: 1
RANGE   1 HIGH LIMIT: 10
RANGE   2 LOW LIMIT: done

THERE ARE    1316 PICTURES PRESENTLY SELECTED

PARAMETER: filter
VALUE   1: red
VALUE   2: green
VALUE   3: done

THERE ARE       0 PICTURES PRESENTLY SELECTED

THEREFORE, NO SCAN MAY BE DONE

ENTER MAIN PROCESSING OPTION: backup

THERE ARE    1316 PICTURES PRESENTLY SELECTED

ENTER MAIN PROCESSING OPTION: search

THERE ARE    1316 PICTURES PRESENTLY SELECTED

PARAMETER: filter
VALUE   1: minus
VALUE   2: done

THERE ARE      30 PICTURES PRESENTLY SELECTED

PARAMETER: qual
VALUE   1: clear
VALUE   2: done

THERE ARE       7 PICTURES PRESENTLY SELECTED

PARAMETER: done

ENTER MAIN PROCESSING OPTION: print
ENTER PRINT REQUEST: micro on

434B01

MICROFICHE RETRIEVAL SECTION
IF YOU DO NOT HAVE AN IMAGE SYSTEMS' MICROFICHE VIEWER, ENTER ABORT.
ELSE ENTER FICHE DEVICE (DEFAULT=TT2:): tt1:
ENTER NEXT COORD: help

YOU ARE IN THE FICHE EDIT SUBROUTINE
THE MICROFICHE VIEWER IS BEING ACCESSED (HOPEFULLY).
    YOU MAY USE THESE COMMANDS: (BESIDES "DONE", "BYE" AND "HELP")
```

```
ADD [PARAMETER]      -PLACES [PARAMETER] IN THE PRINT LIST
CLEAR                -REMOVES ALL PARAMETERS FROM THE PRINT LIST
COPY [FILENAME]      -PARAMETERS IN [FILENAME] ARE PLACED IN THE PRINT LIST
DELETE               -CURRENT PICTURE IS DELETED FROM THE PICTURE LIST
DESCRIBE [PARAMETER]-PRINTS A DESCRIPTION OF [PARAMETER]
DUMP                 -PRINTS PARAMETER VALUES FOR ENTIRE PICT. LIST
MICRO (ON/OFF)       -ENABLE/DISABLE MICROFICHE VIEWER
NEXT                 -SELECTS NEXT PICTURE IN PICTURE LIST AND PRINTS
PREVIOUS             -SELECTS PREVIOUS PICTURE IN PICTURE LIST AND PRINTS
PRINT [+N (OR) -N]   -PRINTS PARAMETER VALUES FOR CURRENT PICTURE (OPTION:+OR-N)
START                -NEXT PICTURE PRINTED WILL BE THE FIRST IN THE PICT. LIST
    NOTE: 1)YOU NEED TYPE JUST THE FIRST 3 LETTERS OF A COMMAND
          2)"START" THEN "PREVIOUS" TYPES LAST PICT. IN LIST
OTHERWISE, THIS ROUTINE WANTS EITHER A VALID FICHE ID
        (SUCH AS 1A1, 2-B-5, AND SO ON)
A VALID COORDINATE ON A FICHE (SUCH AS 1A OR D5)
OR A SET OF COORDINATE EDIT COMMANDS:
        > : INCREMENT COORD LETTER
        < : DECREMENT COORD LETTER
        ^ : INCREMENT COORD NUMBER
        - : DECREMENT COORD NUMBER

ENTER NEXT COORD: next

434B05

ENTER NEXT COORD: prev

434B03

ENTER NEXT COORD: delete

434B05

ENTER NEXT COORD: add res
ENTER NEXT COORD: dump

PICNO     RES
--------------------
434B01    8
434B05    7
434B07    7
434B09    7
434B11    7
434B13    8

END OF LIST REACHED: RESETTING TO START
ENTER NEXT COORD: done
```

10.6. Low Sun Angle Pictures of the Tharsis Plains:

This search was designed to find Viking Orbiter pictures in MCNO 8 (Tharsis

quadrangle) that have incidence angles from 60 to 90 degrees, measured from the normal to the surface, resolutions bewteen 1 and 20 meters/pixel, and good seeing conditions. The use of ABORT is illustrated in an example where the LATLON parameter was entered (for an area search) and the user then decided to alter the search strategy. BACKUP is used after an initial search on the QUAL parameter proved to be to restrictive. QUAL was intentionally entered in response to a main processing option request, to show BIRP's response. DUMP is used in the PRINT option to print the whole picture list.

```
ENTER MAIN PROCESSING OPTION: restore
ENTER MAIN PROCESSING OPTION: search

THERE ARE   39757 PICTURES PRESENTLY SELECTED

PARAMETER: latlon
ENTER AREA SEARCH MODE: point
LONGITUDE (AZIMUTH) OF POINT 1: abort
PARAMETER: target
VALUE   1: mars
VALUE   2: done

THERE ARE   35655 PICTURES PRESENTLY SELECTED

PARAMETER: mcno
VALUE   1: 9
VALUE   2: done

THERE ARE    1544 PICTURES PRESENTLY SELECTED

PARAMETER: inang
RANGE   1 LOW LIMIT: 60
RANGE   1 HIGH LIMIT: 90
RANGE   2 LOW LIMIT: done

THERE ARE    1125 PICTURES PRESENTLY SELECTED

PARAMETER: res
RANGE   1 LOW LIMIT: 1
RANGE   1 HIGH LIMIT: 20
RANGE   2 LOW LIMIT: done

THERE ARE      99 PICTURES PRESENTLY SELECTED

PARAMETER: qual
VALUE   1: clear
VALUE   2: done
```

```
THERE ARE       0 PICTURES PRESENTLY SELECTED

THEREFORE, NO SCAN MAY BE DONE

ENTER MAIN PROCESSING OPTION: backup

THERE ARE      99 PICTURES PRESENTLY SELECTED

ENTER MAIN PROCESSING OPTION: qual

I DON'T UNDERSTAND--TYPE 'HELP' FOR HELP.

ENTER MAIN PROCESSING OPTION: search

THERE ARE      99 PICTURES PRESENTLY SELECTED

PARAMETER: qual
VALUE   1: clear
VALUE   2: slight
VALUE   3: done

THERE ARE      27 PICTURES PRESENTLY SELECTED

PARAMETER: done
ENTER MAIN PROCESSING OPTION: print
ENTER PRINT REQUEST: add inang
ENTER PRINT REQUEST: add res
ENTER PRINT REQUEST: add clat
ENTER PRINT REQUEST: add clong
ENTER PRINT REQUEST: dump
```

PICNO	INANG	RES	CLAT	CLONG
360A31	65.30	17	18.44	133.80
360A32	65.20	17	18.62	133.52
360A33	65.10	17	18.34	133.66
360A34	65.04	17	18.52	133.42
360A35	64.92	17	18.24	133.52
360A36	64.86	17	18.44	133.28
360A37	64.74	17	18.14	133.42
360A38	64.70	18	18.36	133.18
360A39	64.60	18	18.06	133.32
360A40	64.56	18	18.28	133.06
360A41	64.46	18	17.98	133.22
360A42	64.44	18	18.22	133.02
360A43	64.34	18	17.92	133.14
360A44	64.32	19	18.16	132.94
360A45	64.24	19	17.88	133.12
360A46	64.24	19	18.12	132.88
360A47	64.16	19	17.84	133.06
360A48	64.18	20	18.10	132.88
360A49	64.10	20	17.80	133.04
360A50	64.14	20	18.08	132.86
387A01	71.10	18	10.72	104.88

```
387A02    70.90    19        10.86     104.60
387A03    70.70    19        10.44     104.60
387A04    70.48    19        10.58     104.30
387A05    70.28    19        10.14     104.32
387A06    70.06    19        10.30     104.00
387A07    69.86    19         9.86     104.04

END OF LIST REACHED: RESETTING TO START
ENTER PRINT REQUEST:done
```

10.7. Limb Shots With Red Filter During Dust Storms:

This search is designed to find those limb frames taken through the red filter during the two Martian dust storms that occurred during the first Martian year of Viking operations. The subsolar longitude (LS) parameter was used to restrict the picture list to those pictures taken with LS values between 208 to 345 degrees. The 2 global dust storms are bracketed by this range. The QUAL parameter was used to insure that the pictures selected were taken under hazy conditions. Use of copying a standard file containing engineering parameter names to be included in the print list is illustrated.

```
ENTER MAIN PROCESSING OPTION: restore
ENTER MAIN PROCESSING OPTION: search

THERE ARE   39757 PICTURES PRESENTLY SELECTED

PARAMETER: target
VALUE   1: limb
VALUE   2: done

THERE ARE    1311 PICTURES PRESENTLY SELECTED

PARAMETER: ls
RANGE   1 LOW LIMIT: 208
RANGE   1 HIGH LIMIT: 345
RANGE   2 LOW LIMIT: done

THERE ARE     494 PICTURES PRESENTLY SELECTED

PARAMETER: filter
VALUE   1: red
VALUE   2: done

THERE ARE     209 PICTURES PRESENTLY SELECTED
```

```
PARAMETER: qual
VALUE   1: dense
VALUE   2: done

THERE ARE      13 PICTURES PRESENTLY SELECTED

PARAMETER: done
ENTER MAIN PROCESSING OPTION: print
ENTER PRINT REQUEST: copy standard.pri
ENTER PRINT REQUEST: dump
```

PICNO	CLAT	CLONG	PICHT	PICWD	RES	SLANTR	QUAL
	LS	FILTER	INANG	EMANG	PHANG	TARGET	
369A01	21.26	312.82	769.10	-1929.40	677	27092.0	DENSE
	284.04	RED	70.74	52.60	55.90	LIMB	
369A31	-78.12	138.26	783.10	-3145.00	683	27348.0	DENSE
	284.04	RED	73.42	94.62	55.88	LIMB	
369A35	22.58	325.54	1080.30	-1796.30	687	27484.0	DENSE
	284.04	RED	63.48	55.52	57.50	LIMB	
369A37	9.78	324.14	841.60	1521.70	670	26828.0	DENSE
	284.04	RED	55.98	46.44	57.58	LIMB	
369A63	-78.66	80.52	939.50	3157.10	683	27356.0	DENSE
	284.04	RED	62.30	94.48	57.48	LIMB	
369A67	23.50	337.99	3156.70	-1657.90	698	27940.0	DENSE
	284.04	RED	57.46	59.24	58.90	LIMB	
369A69	8.26	334.14	1017.50	1559.30	678	27120.0	DENSE
	284.04	RED	48.32	49.24	58.92	LIMB	
368A93	-69.90	51.26	1223.20	3213.60	687	27484.0	DENSE
	284.04	RED	51.16	94.68	59.08	LIMB	
385A12	-76.38	248.24	1056.10	1664.00	752	30092.0	DENSE
	293.38	RED	62.44	74.92	85.62	LIMB	
312B20	63.56	185.60	330.40	1099.80	302	12118.0	DENSE
	289.40	RED	87.24	90.74	71.30	LIMB	
315B40	57.32	127.58	413.90	1369.60	366	14674.0	DENSE
	291.14	RED	82.00	90.54	68.06	LIMB	
391B30	-13.14	47.44	1422.00	1379.40	683	27346.0	DENSE
	332.62	RED	37.04	79.92	88.80	LIMB	
391B39	-53.72	353.86	1622.40	1472.10	678	27132.0	DENSE
	332.62	RED	44.40	55.68	88.12	LIMB	

```
END OF LIST REACHED: RESETTING TO START
ENTER PRINT REQUEST:done
```

10.8. Seasonal Variations in a Crater Streak in the Elysium Quadrangle:

This search is designed to find pictures over a crater in the Elysium quadrangle, at 191 deg. W. longitude, 7 deg. N. latitude, that might show seasonal variations in brightness or color. First, MCNO was used to restrict frames of Mars

to those located at or near the crater. Then LATLON was used for a point area search. Only pictures that have some part of their area overlying the point are included in the picture list after this search. SAVE was then used to save the bit map file from this search after first printing the parameter values. The program CREATE was used to change the file into a BIRP parallel data file that can be accessed in future searches by use of the CRASTK parameter name. The RSX-11M MCR command BYE is then used to log-off from the ORBITER account.

```
ENTER MAIN PROCESSING OPTION: restore
ENTER MAIN PROCESSING OPTION: search

THERE ARE   39757 PICTURES PRESENTLY SELECTED

PARAMETER: target
VALUE   1: mars
VALUE   2: done

THERE ARE   35655 PICTURES PRESENTLY SELECTED

PARAMETER: mcno
VALUE   1: 8
VALUE   2: 16
VALUE   3: 15
VALUE   4: 23
VALUE   5: done

THERE ARE    3472 PICTURES PRESENTLY SELECTED

PARAMETER: latlon
ENTER AREA SEARCH MODE: point
LONGITUDE (AZIMUTH) OR POINT 1: 191
LATITUDE (ELEVATION) OF POINT 1: 7

THERE ARE      24 PICTURES PRESENTLY SELECTED

PARAMETER: done
ENTER MAIN PROCESSING OPTION: print
ENTER PRINT REQUEST: add filter
ENTER PRINT REQUEST: add phang
ENTER PRINT REQUEST: add res
ENTER PRINT REQUEST: dump

PICNO     FILTER    PHANG     RES
---------------------------------------
409A01    RED       32.54     405
421A31    VIOLET    56.20     835
421A33    CLEAR     56.20     835
```

```
421A35    CLEAR     56.20     835
421A37    GREEN     56.20     835
421A39    RED       56.18     835
506A04    VIOLET    38.14     601
506A24    GREEN     38.62     595
506A26    GREEN     39.84     597
605A28    RED       29.56     853
635A51    RED       93.88     259
635A52    RED       94.90     263
672A61    RED       98.26     326
684A21    VIOLET    55.10     828
684A23    CLEAR     55.12     828
684A25    GREEN     55.14     828
684A27    RED       55.14     828
687A41    VIOLET    54.70     821
687A43    CLEAR     54.72     821
687A45    GREEN     54.72     821
687A47    RED       54.72     821
747A73    RED       111.40    537
747A76    RED       110.36    534
747A85    RED       103.70    646

END OF LIST REACHED: RESETTING TO START

ENTER PRINT REQUEST: done
ENTER MAIN PROCESSING OPTION: save
ENTER NEW PARAMETER NAME FOR LIST: crastk

WE SHALL NOW EXIT TO THE OPERATING SYSTEM,
TO ALLOW YOU TO EDIT THE DESCRIPTION FILE FOR YOUR
NEW PARAMETER FILE: DO THE FOLLOWING:

EDI CRASTK.TXT      (WHICH STARTS THE EDITOR)
 < ENTER THE DESCRIPTION OF THE PARAMETER, TERMINATED BY
 A LINE WITH NOTHING ON IT>
EX      (WHICH EXITS FROM THE EDITOR)
RUN CREATE    (TO CREATE THE BIRP DATA FILE)
CRASTK   TO TELL CREATE WHAT FILE TO MAKE
DONE  (TO LEAVE CREATE)
 AND RUN BIRP TO GET BACK INTO BIRP.


>edi crastk.txt
[CREATING NEW FILE]
INPUT
crastk-results of seasonal search streak at 171w,9n

*ex
[EXIT]

>run create
ENTER PARAMETER NAME TO CREATE: crastk
ENTER PARAMETER NAME TO CREATE: done

>bye
```

```
>
HAVE A GOOD AFTERNOON
20-NOV-79 12:27  TTO:   LOGGED OFF
```

11. AUTOGEN

The original source data of a database will rarely be in the parallel parameter file format BIRP requires. More typicaly, the original data are organized so that all the information about a given picture (or other single unit) is on one record. This is standard for the Viking Orbiter data tapes for example. AUTOGEN and CREATE are two programs written to assist the user to disassemble large quantities of source data and re-format those data into BIRP standard data files. AUTOGEN requests user input for the size and location of the parameters to be turned into PARAMETER.BRP files and it then steps through the original source data and extracts the proper data. Then, with the aid of a text file made by the user, CREATE assembles the final PARAMETER.BRP file for each parameter found useful.

11.1. SYSTEM CONSIDERATIONS

The process of disassembling the orginal source data can be quite complex. For example, DEC PDP-11 computers store integers in a opposite form than IBM computers. In addition, the IBM character code (EBCDIC) is incompatible with DEC computers. AUTOGEN is designed to deal with DEC formats only. Any re-formating of data to DEC standard must be done first. A sample program for this purpose, PREAUTOGEN, can be obtained from the authors. It has ample comments to guide the user and may be a good starting point to construct a pre-processing program that meets the user's own particular problems in data re-formating.

11.2. DATA STRUCTURES FOR AUTOGEN

There are three data structures that the user must be familiar with to use AUTOGEN: (1) The original source database, (2) The parameter data file produced by AUTOGEN (PNAME.DAT), and (3) The parameter identification file (PNAME.IDX).

As discussed, the orginal source database must be DEC PDP-11 compatible. This database must be (1) A direct access file on disk, (2) Formatted so that each record contains all the data for a particular picture or other similar data unit and (3) Text data coded in ASCII, and other data formated as per the FORTRAN IV+ User's manual (manual no. EC-11-LFPUA-B-D, Appendix A).

There is one PNAME.DAT file built for each PARAMETER.BRP file used in a BIRP database library. The PNAME.DAT file is the data area of the PARAMETER.BRP file. After CREATE constructs the header area it then copies the PNAME.DAT file directly into the PARAMETER.BRP data area. All numerical data more then 1 byte long are stored by AUTOGEN in the PNAME.DAT file in "scrambled" (MSB first) form.

As with PNAME.DAT, there is one PNAME.IDX file per PARAMETER.BRP file. PNAME.IDX is a text file, created by AUTOGEN, but easily modified or corrected by a text editor. The file is sequential access. All numerical fields may be terminated by a comma. AUTOGEN puts the standard name for the data on (example: I/O DATA TYPE) each line after the comma. The structure of the PNAME.IDX file is shown in Table 11.1.

Note that these entries are the same as those in the PARAMETER.BRP header; CREATE merely re-formats the PNAME.IDX data and places that information at the beginning of the PARAMETER.BRP file that it is currently building. The PNAME.IDX file gives the user additional flexibility in correcting errors from AUTOGEN discovered after running BIRP using a

PARAMETER.BRP file. As long as the PNAME.DAT file has been preserved, the user may construct a new PNAME.IDX file with a text editor and run CREATE.

11.3. RUNNING AUTOGEN

AUTOGEN is a stand-alone program, built by the user together with two subroutines: GREQ and GIREQ. AUTOGEN contains HELP messages that are printed for each query. Below is a list of AUTOGEN questions in proper order, with a short explanation for each.

(1) ENTER THE NUMBER OF PICTURES IN THE LIBRARY:

This question is only asked when no BIRP.BRP file with the number of pictures in the library exists. When AUTOGEN is run again on the same account, this question will not be asked, since the BIRP.BRP file would have been generated during the first run of AUTOGEN. Maximum number of pictures or records is 9,999,999.

(2) ENTER MAIN FILE NAME:

(3) ENTER THE NUMBER OF BYTES/RECORD:

This is the complete name (in RSX-11M = BP1:[group,member]NAME.EXT) of the original data file that the PNAME.DAT files will be a extracted from. This file must be direct access file on the user's account. The length in bytes of each record on this file (data for each picture) must be entered, with a maximum size being 32,767 bytes.

(4) ENTER NEXT PARAMETER NAME:

This is the PNAME for the PNAME.DAT file to be built on a given pass through AUTOGEN. Maximum length of text is 9 characters.

(5) ENTER START DATA BYTE LOCATION:

Autogen assumes that a particular datum is always in the same relative position in each record of the original data file. Therefore, the answer to this question is the number of bytes from the beginning of each record to where the datum, which is to be placed in the PNAME.DAT file, is located. The maximum allowed is the same number as the record size entered in question 3.

(6) ENTER WIDTH CODE:

This is the same code detailed in the discussion of the PARAMETER.BRP structure (section 3.1). The code is the length of the datum to be extracted from the original data file. The code may not be a 0, which is the width code that corresponds to bit type data. Bit type data may only be constructed by running BIRP, then saving a bit map, then running CREATE with the PNAME.DAT and PNAME.IDX files built by the SAVE option. Thus, the legal values for width code and their actual corresponding data lenghts are: 1 (1 byte), 2 (2 bytes), 3 (4 bytes), 4 (8 bytes), 5 (16 bytes), 6 (32 bytes) and 7 (64 bytes). Clearly the user may be forced to put the original data through some form of proccessing to fit these required lengths.

(7) I NEED THE EXTRA INFORMATION CODE:

The extra information code allows BIRP to handle unusual data types which may at some time require special processing. At present only 0 = standard data, 1 = latitude and 2 = longitude are used. There may be up to 31 different extra information types, if these are ever needed.

(8) ENTER I/O TYPE

Bit type files are not a valid I/O type to be entered in response to question 6. A bit type file can only be produced by running BIRP and using the SAVE option. So, the permissible values for I/O types are: 2 (byte, keyed by text list), 3 (integer), 4 (scaled integer), 5 (text) and 6 (floating point number).

(9) ENTER THE REMAINING TEXT LINES AS THEY SHALL APPEAR IN THE INDEX:

The index referred to is the one derived from the PARAMETER.BRP header area (section 3.3). If the I/O type is 2 a list of text keys is needed, if the I/O type is 4 a real scale factor is required. Those entries are made at this point. Up to 30 text keys, one per line, each up to 8 characters long may be used. The numerical value of the data determines which text key is used on output: 0 = first key, 1 = second key and so on. When a scaled integer I/O type is used, a floating point scale factor is entered at this point. This scale factor will be read into BIRP using a F16.0 format. The text keys are typed in with a carriage return after each; the scale factor takes just one line. When all text keys or the scale factor are typed in, type DONE on the next line and AUTOGEN will begin processing the original database file to produce the single PNAME.DAT and PNAME.IDX files described by the entries made.

11.4. PROGRAM FLOW THROUGH AUTOGEN

AUTOGEN uses two subroutines: GIREQ and GREQ, to ask questions and process (in almost all cases) the answers, whether numeric or text. GIREQ accepts numeric user input. GIREQ is called with a range of acceptable values and a prompt number, which is passed to GREQ to type out the proper

question. GIREQ tests if the user's reply is within proper range; if not it prints the actual acceptable range to assist the user. When text answers are required, GREQ is called directly from AUTOGEN and prints a prompt. GREQ returns the user's text reply in a 80 character buffer.

AUTOGEN starts by opening a BIRP.BRP file on the user's account (with DEC FORTRAN IV+ OPEN statement keyword TYPE = 'UNKNOWN') and checking for an integer number in the first record. If the file did not exist before the open, there will not be a number. AUTOGEN then prompts GIREQ to ask for the number of pictures on the library. After AUTOGEN is finished the BIRP.BRP file may be edited to add more textual information. This text is printed in BIRP when the user types the main processing option DESCRIBE. AUTOGEN goes on to write the number input by the user in the newly created BIRP.BRP file, along with the text: 'DUMMY LIBRARY' in the second record (to be replaced by the user with proper name).

After asking the user question 2, AUTOGEN attempts to open the original database file as a direct access file. Then, with the answer to question 4, the PNAME.IDX file is opened as a sequential file. After question 6, AUTOGEN tests, using the start byte, width code and bytes/record figures entered by the user, whether the end of the datum to be accessed in the original database file will extend beyond the end of a record. After question 9, (conversion text), AUTOGEN writes the information input by the user to the PNAME.IDX file. Then the text lines entered in response to question 10, are read one by one to the PNAME.IDX file until the user types DONE.

AUTOGEN now calculates, using the number of pictures and the width code, the size of the PNAME.DAT file. This file always occupies an integral number of blocks (512 bytes). When BIRP scans the PARAMETER.BRP file data area, data is brought into memory block by block rather then datum by datum, to

reduce the number of disk accesses needed. In the same way AUTOGEN now gets the each datum value from the original database file and places it in a 512 byte buffer in sequential fashion. Each datum is reversed (to MSB first order) by the routine SCRMBL. When the buffer is filled, DMPBYT writes it to the PNAME.DAT file. This continues until all datum values have been transfered to the PNAME.DAT file. Now AUTOGEN requests a new parameter name (question 4) from the user and the process repeats.

12. CREATE

CREATE is a stand alone program that builds the PARAMETER.BRP files from three intermediate input files. The three input files are PARAMETER.DAT, PARAMETER.IDX and PARAMETER.TXT, and each are discussed in section 11. The PARAMTER.DAT file contains the parameter values for all of the pictures in the data set. The PARAMETER.IDX contains the information needed to create the header area of the PARAMETER.BRP file (see section 3.1.1). The PARAMETER.DAT and PARAMETER.IDX files are usually created either by AUTOGEN for new data sets or by the save option of BIRP. The PARAMETER.TXT file contains text that describes the new parameter file. It is created by the system editor. When CREATE is run, a parameter name is requested. Then the program attempts to find the .DAT, .IDX and .TXT files with the parameter name input by the user. If any of these files are not present, then an error is reported and a new name is requested. Otherwise, a PARAMETER.BRP file is created.

CREATE first reads the PARAMETER.IDX file into memory and checks that the entries have legal values. The PARAMETER.TXT file is then opened and its length is determined. CREATE then builds the header area for the new PARAMETER.BRP file. Finally, the data from the PARAMETER.DAT file is

transferred to the PARAMETER.BRF file. The new parameter file is now finished and ready to use.

APPENDIX I

BUILDING BIRP ON YOUR SYSTEM

Systems Running RSX-11M

If your BIRP source is a DOS-format magtape, the indirect command file FLXBRP.CMD on account [220,1] may first be transferred to disk. The indirect command file can then be used to bring the rest of the BIRP system and the Viking Orbiter and Lander files (at present) over to the logical disk BP1:. There are tasks on account [220,1] which should run on most RSX systems above 3.0 when the task images are made contiguous. These tasks are BIRP.TSK, CREATE.TSK and AUTOGEN.TSK. If these tasks must be rebuilt for any reason, indirect command files BRPTKB.CMD, CREATETKB.CMD and AUTOTKBL2.CMD will be helpful. The indirect command files may be examined and edited if certain options or switches are different on the user's system. All the object modules are included on the account [220,1]. The object modules were compiled with DEC PDP-11 FORTRAN IV+. BRPCMD.CMD automatically compiles all the BIRP routines properly and then links to BRPTKB.CMD, if the user wishes.

RSX-11M with DEC FORTRAN IV

The two present DEC PDP-11 FORTRANs, IV and IV+ have several basic incompatibilities. The most important of these is FORTRAN IV's lack of an I*4 data type. Most of the other differences: (depending on the FORTRAN IV version): 1) Entry points, 2) lower bounds in array declarations, 3) generic function selection by argument data type and 4) certain differences in the OPEN statement, may be programmed around without great difficulty. However, many subroutines use the number of pictures in the accessed library (usually called NPICTS) as an argument. Of course this must be an integer, and in the

example of the Viking Orbiter, the value exceeds the range of an I*2 variable. Thus, NPICTS must be an I*4 type. Two strategies might be explored to circumvent this problem: 1) A macro routine could be written to handle the two words passed in the argument and seperate them into two I*2 variables if any processing is neccessary in the subroutine, or 2) The argument might be split into two arguments in the calling sequence, so the number of pictures on the library would always be stored in two I*2 variables. It should be noted that very few computational languages on medium-sized and larger systems have the severe restriction on integer size of DEC FORTRAN IV.

BIRP on non-PDP-11 Systems

The DEC FORTRAN IV+ programming language is very similar to the ANSI FORTRAN 77 standard. However, BIRP will probably require some modifications in order to run on non-PDP-11 systems. The only important difference between DEC FORTRAN IV+, as applied to BIRP, and ANSI FORTRAN 77 is in the use of byte arrays. The following changes must be considered when modifying BIRP to run on systems that do not have a byte data type, with 8 bits in each byte. The nature of the changes is strongly system dependent and, as a consequence, we can only point out possible problem areas.

(1) Many BIRP routines use bitwise logical functions (IAND, ISHIFT and IEOR), which are intrinsic DEC FORTRAN IV+ functions. If the system does not have such logical functions, the equivalent in FORTRAN or assembly language code will have to be provided.

(2) BIRP, through the subroutine SCRMBL, swaps bytes to insure that the MSB is first within a word. If your system already has the MSB first, such byte swapping will be unnecessary.

(3) BIRP is heavily dependent on the use of byte arrays. Byte data types are not available in FORTRAN 77. In many cases changing the byte variables to integers should work. The following items will require additional changes:

(a) Text for input and output is stored in byte arrays, with one character in each array element.

(b) The index buffer and scan control table (see section 2) are byte arrays which store a mixture of data types.

(c) Parameter data, regardless of the data type, are read from the PARAMETER.BRP files into a byte array. The routines CVTIN and CVTOUT, which convert the byte arrays to the proper data type for input and output, will have to be changed.

(d) The routines INALID and SETSCN assume that there are 8 bits in a byte.

(e) The assembly language routines CPYBYT and CMPARB require that the bytes within words can be individually addressed.

The use of assembly code is limited and well documented in the source files, although the systems programmer may want to consult a copy of the PDP-11 Macro programming manual.

APPENDIX II

DESCRIPTION OF BIRP ROUTINES

All of the routines in the BIRP system, including AUTOGEN and CREATE are listed below. A brief description of the purpose of the routines is also provided. Refer to the source code for details of program flow, etc.

AREFLD - REQUEST AREA FIELD VALUES FROM USER
AREFLD requests area search parameters from the user. There are two possible area search modes; point and area. AREFLD sets up the search scan control table, which directs the search routine.

ARESCN - AREA TYPE SCAN OF DATA FILE
ARESCN scans the area data files to find images that overlap with the user defined area or point.

ASCBFC - ASCII TEXT TO BINARY FICHE CONVERT
ASCBFC is used in the microfiche section of BIRP. It converts the formatted ASCII equivalent of a microfiche clip code into the binary code (one word) form.

AUTOGEN - AUTOMATIC DATA FILE GENERATION
AUTOGEN is a stand alone program which takes, as input, the engineering records describing a given picture in a given mission. The program extracts those parameters important to a user and places the values in a set of parallel data files, ordered by picture sequence number.

BFCASC - BINARY FICHE CLIP CODE TO ASCII TEXT CONVERT
BFSASC is used in the microfiche section of BIRP. It converts a binary clip code (one word) to the ASCII code (3 characters) on the viewer front panel.

BINFCH - BINARY TO FICHE CLIP CODE CONVERT
BINFCH is used in the microfiche section of BIRP. It converts a binary form of the clip code to the command code that is sent to the microfiche viewer, which select a particular fiche card.

BINFLC - BINARY TO FICHE CARD LOCATION CONVERT
BINFLC is used in the microfiche section of BIRP. It converts a binary row-column address of the frame on a card to a command code understood by the microfiche reader.

BIRP - BETTER IMAGE RETRIEVAL PROGRAMS
BIRP is the main calling program. Based on user inputs, it selects one of the six main processing options: describe, print, search, save, restore and backup.

BLCASC - BINARY FICHE CARD LOCATION TO ASCII TEXT CONVERT
BLCASC is used in the microfiche section of BIRP. It converts a binary row-column address of a single frame within a microfiche card to the two

character ASCII front panel form (e.g., A1, L5).

BOX - BOX TYPE AREA SEARCH
BOX performs the preprocessing for the area search routines. It tests whether or not a picture crosses a latitude band defined by the maximum and minimum latitudes of the user defined search area.

CLRPID - CLEAR PRESENTLY SELECTED PICTURE ID
CLRPID is part of SETSCN. It removes a picture from the list of selected pictures by clearing the bit in the bit map representing that picture ID.

CMPARB - COMPARE BYTE BUFFER UTILITY
CMPARB compares two byte buffers and indicates whether the buffers match exactly or not. If the buffers are not equal, CMPARB indicates which buffer had the first non-matching byte that was larger. This routine is useful for comparing both text and numerical data.

CVTIN - CONVERT INPUT STRING TO BIRP FORM
CVTIN converts a user's input of a search constraint into a BIRP internal format.

CPYBYT - COPY BYTE BUFFER UTILITY
CPYBYT copies the contents of one buffer into another buffer, byte by byte, given the starting address of each buffer and the number of bytes to be transferred.

CREATE - CREATE BIRP DATA FILES
CREATE is a stand alone program which produces the final PARAMETER.BRP data files required by BIRP. The files input to create are the PARAMETER.DAT, PARAMETER.IDX and PARAMETER.TXT. These files come from either AUTOGEN or the save option of BIRP. The PARAMETER.BRP file contains all the information BIRP needs to access or search that data file.

CVTOUT - CONVERT BIRP FROM TO OUTPUT STRING
CVTOUT takes data in the BIRP internal format used and converts it to an ASCII character string, so that it may be output on the user's terminal.

DMPBYT - DUMP SEQUENTIAL DATA BYTES TO CONTIGUOUS OUTPUT FILE
DMPBYT writes a buffer to a disk file in blocks that are 512 bytes long, given the starting address of the buffer and its length. This block length matches the disk block size and thus speeds data transfers.

FCHEDT - MICROFICHE EDIT
FCHEDT allows the user to select any microfiche card by its clip code and any frame within a given card by moving up or down one row and left or right one column.

FCHLIB - MICROFICHE CONVERSION SUBROUTINE LIBRARY
FCHLIB is the module that contains the library of microfiche conversion utilities. It contains ASCBFC, ASCBLC, BFCASC, BINFCH, BINFLC and BLCASC.

FINSCN - FINISH SCANNING THE BIT MAP
FINSCN is part of the INALID routines. Its purpose is to write the last sector of the bit map from the computer memory to a disk file.

FNDSPE - FIND SPECIAL CHARACTERS IN TEXT BUFFER
FNDSPE searches a text buffer byte by byte for a match with the characters specified by the calling routine. FNDSPE returns the location of the first special character. A maximum location for the input text buffer may be specified so that a match will not be sought beyond that location.

FNDVAL - FIND VALUE IN THE SCAN CONTROL TABLE
FNDVAL is used by REQFLD to examine the scan control table. The scan control table contains a list of user search criteria for a particular parameter. FNDVAL compares a new user input to those already in the table, and indicates the location of the first existing value larger than the input of the user.

GETBVL - GET BIRP DATA VALUE FROM DATA FILE
GETBVL is given a picture ID number (the number that corresponds to the position of desired picture within the BIRP parameter files). GETBVL finds and returns the value of that picture from a specific parameter file.

GETBYT - GET DATA BYTE FROM FILE
GETBYT is used to read data from parameter files. It reads a full disk sector (512 bytes) only if the requested data is not in the internal buffer of GETBYT.

GETNBT - GET NEW RECORD DATA BYTE
GETNBT is a entry point of GETBYT which always does a read from disk. It assumes that the requested data is not in the GETBYT buffer.

GIREQ - GET INTEGER REQUEST FROM USER FOR AUTOGEN
GIREQ is used by AUTOGEN to process numerical inputs. GIREQ is given an upper and lower limit for the user's input and indicates whether the user's input is within this range.

GNIDBT - GET NEXT ID BYTE FROM BIT MAP
GNIDBT is part of the INALID library. GNIDBT gets the next byte of the bit map, either from a memory buffer or from disk, if necessary. It checks whether the end of the bit map has been encountered.

GREQ- GET EDIT REQUEST FROM USER FOR AUTOGEN
GREQ is used by AUTOGEN. It asks the user for certain information about a specific parameter such as type and size of data.

GNXPID -GET NEXT PICTURE ID FROM BIT MAP
GNXPID is part of SETSCN. GNXPID searches the bit map for the next bit that equals 1 which corresponds to the next picture still selected (not yet eliminated by a search). The position of that picture is the data files in returned by GNXPID.

INALID - INITIALIZE ALL PICTURE ID'S

INALID sets up the two bit maps as scratch disk files. All the bits in the bit maps are set to 1 (all pictures are available for a search). INALID has multiple entry points. The purpose of the routines in INALID is to manipulate the bit maps.

INIBPT - INITIALIZE BIT MAP POINTERS

INIBPT is a part of INALID. INIBPT resets internal pointers so that the next byte access will be the first byte in the bit map.

I4DMP - INTEGER*4 DATA DUMP TO FILE

I4DMP is used by CREATE. It calls DMPBYT to write an INTEGER*4 variable to the PARAMETER.BRP data file in the internal BIRP format (i.e., the most significant byte first).

LSTRND - LIST TYPE SCAN OF RANDOM DATA FILES

LSTRND scans a parameter data file, whose data values are in random order, to find matches between items in the file and a list of user specified values. Only pictures with matching data remain selected for subsequent searches or output.

LSTSEQ - LIST TYPE SCAN OF SEQUENTIAL DATA FILES

LSTSEQ scans a parameter data file, whose data values are in numerical order, to find matches between items in the file and a list of user specified values. Only pictures with matching data are selected for future searches or output.

MICGO - MICROFICHE EDIT CONTROL, GO

MICGO is called by PRTBRP and controls all access to the microfiche viewer. MICGO uses the FCHEDT and FCHLIB routines.

NTRGLE - NORTH TRIANGLE ROUTINE

NTRGLE is used by QUAD to calculate the bearing from north of a line that connects two points on a sphere. NTRGLE also determines the cord length between the two points.

OPNBRP - OPEN BIRP DESCRIPTION FILE

OPNBRP opens the BIRP.BRP file and prints on the user's terminal the total number of pictures available in that dataset and the name of the dataset. If requested OPNBRP will print the rest of the BIRP.BRP file which contains descriptions of the parameters in the dataset.

OPNPRM - OPEN BIRP PARAMETER DATA FILES

OPNPRM reads the index section of the PARAMETER.BRP files, which contains information about the size and type of data in that file. OPNPRM then checks this index for proper format and internal consistancy.

PRTBRP - PRINT BIRP DATA FILE VALUES

PRTBRP directs the output of parameter values for pictures selected by the search routines. It also allows the pictures to be viewed on the microfiche machine.

PPIDBT - PUT PRESENT ID BYTE INTO BIT MAP
PPIDBT is a part of INALID. It places the value of a byte variable that was passed to it in the location indicated by the bit map pointer.

PUTSVL - PUT SCAN VALUE INTO THE SCAN CONTROL TABLE
PUTSVL takes a user supplied search value and places it in the scan control table. The new data is placed in the location determined by FNDVAL.

QUAD - FIND POINT IN QUADRILATERAL
QUAD is used by ACESCN. It determines whether a given point on a sphere is within a quadrilateral defined by 4 other points on the sphere.

REQFLD - REQUEST FIELD VALUES FROM USER
REQFLD takes a series of user supplied search values or ranges and sets up a scan control table. The scan control table is used by the search routines.

REQPRM - REQUEST PARAMETER NAME FROM USER
REQPRM asks the user for the name of a parameter to search. It opens the parameter file and creates the parameter index via OPNPRM.

REQSET - REQUEST SET-UP FOR PARAMETER SEARCH
REQSET requests a parameter name through REQPRM. It then calls REQFLD or AREFLD to set up the scan control table so that the requested parameter file can be searched.

REQUST - TRANSLATE USER REQUESTS FOR BIRP ROUTINES
REQUST handles all inputs from the user. It will print a prompt message, if the calling routine provides one, and accepts a user input. It checks for any of the following specical responses: done, abort, exit, bye, off and help. REQUST returns the user text input and indicates which special response, if any, was made.

RNGRND - RANGE TYPE SCAN OF RANDOM DATA FILE
RNGRND scans a parameter data file, whose data values are in random order, to find all pictures that have parameter values between a series of ranges selected by the user. Only those pictures with data in the specified ranges remain selected for future searches or output.

RNGSEQ - RANGE TYPE SCAN OF SEQUENTIAL DATA FILE
RNGSEQ scans a parameter data file, whose data values are in numerical order, to find all pictures wiht values between a set of ranges selected by the user. Only those pictures with data in the specified ranges remain selected for future searches or output.

SAVBRP - SAVE BIRP BIT MAP
SAVBRP creates a PARAMETER.DAT file out of the current bit map. It lets the user save a particular set of pictures as a new BIRP data file. CREATE has to be run to build the final PARAMETER.BRP file.

SCNBRP -SCAN BIRP DATA FILES
SCNBRP calls the routines that are necessary to search a BIRP data file. It tells the user after each search the number of pictures still selected (not yet eliminated).

SCRMBL - SCRAMBLE DATA TO AND FROM BIRP FORMAT
SCRMBL reformats data for use by BIRP. Within a word, DEC software stores bytes in the reverse manner relative to most other software. BIRP requires data with the most significant byte first within a word. SCRMBL does the required byte swapping depending on the data type.

SETSCN - SET UP SCAN OB BIT MAP
SETSCN sets the bit map pointers to the beginning of the bit map. This readys BIRP for the next search operation.

SHFVLS - SHIFT SCAN VALUES AROUND IN SCAN CONTROL TABLE
SHFVLS shifts the entries in the scan control table to make room for a new entry. This is done so that the entries in the scan control table are always in numerical order.

SKPSEP - SKIP SEPARATORS IN TEXT BUFFER
SKPSEP examines a string of ASCII text. It returns the location of the first character that is not a blank or a comma.

STRSCN - START SCAN OF BIRP FILES
STRSCN selects the proper search routine (LSTSEQ, LSTRND, RNGSEQ, RNGRND, BITSCN or ARESCN) based on the data type of the parameter file that is going to be searched.

SWAPBT - SWAP BYTES IN A WORD
SWAPBT reverses the locations of the two bytes within a single word.

VIEW - MICROFICHE VIEWING CONTROL
VIEW uses the routines in FCHLIB to select the desired microfiche card and frame location for the microfiche viewer.

APPENDIX III

ABBREVIATED GUIDE TO BIRP COMMANDS

GENERAL COMMANDS: MAY BE USED ANYTIME IN BIRP

- HELP - Gives available commands at any point in BIRP.
- BYE, EXIT, OFF - To leave BIRP.
- ABORT - Returns user to previous option if incorrect response is made.
- DONE - 1) Return to main processing option;
  2) In search option: Begin processing after value/range entry.

MAIN PROCESSING OPTION COMMANDS

@ DESCRIBE - Prints descriptions of all parameters in the library being accessed

@ RESTORE - Returns to the condition with all pictures in the library available for searching.

@ BACKUP - Returns to the set of available pictures that was present before the last search.

@ SAVE - Makes the set of pictures now selected a separate parameter file so it may be accessed later.

@ SEARCH - This option asks the user for a specific engineering parameter and a value, a set of values, a range, a set of ranges, or yes/no entry. Then after entering DONE, BIRP will eliminate all pictures which do not meet the user's requirements. In SEARCH, "DESCRIBE <PARAMETER>" prints out a description of <PARAMETER>.

@ PRINT - 1) Prints any or all data from parameter files for pictures selected by searches.
2) Allows viewing of microfiche images of pictures selected.

+ PRINT <N> - Prints data for N pictures from the list of those selected. +N goes forward through the list, -N backwards. Default for N is +1.

+ NEXT - Prints data for next picture in the list of those selected. Also displays microfiche image of this picture if microfiche reader is enabled.

- PREVIOUS - Prints data for previous picture in the list of those selected. Also displays microfiche image of this picture if microfiche reader is enabled.

- START - Goes to first picture in the list of those selected. (use of "PREVIOUS" or "PRINT -1" commands after a START command would go to the end of the list)

- DUMP - Prints all parameters "ADDED" for all pictures selected by searches.

- ADD <PARAMETER> - Adds an engineering parameter to the information included in data listing. Default is picture number.

- COPY <FILENAME> - Engineering parameters in the file "FILENAME" are included in the list of parameter values. Equivalent to a series of "ADD" <PARAMETER>.

- CLEAR - Removes all engineering parameters from the list of parameter values, leaving only picture number.

- DELETE - Removes last picture printed from the list of selected pictures.

- MICRO ON/OFF - Enables/disables automatic display of pictures on microfiche reader.

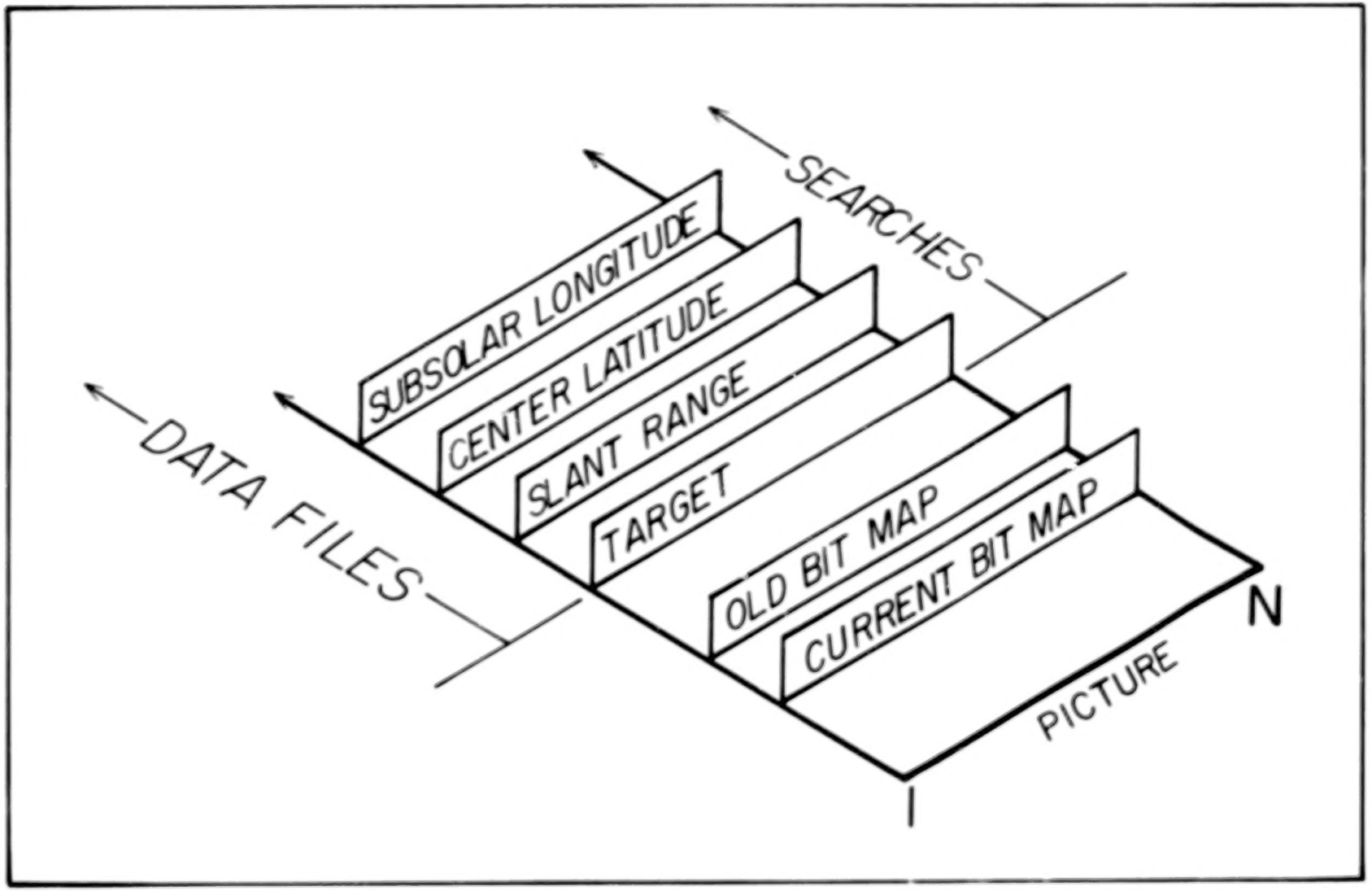
NOTE: After entering MICRO ON, the user also has option to: 1) enter valid fiche ID (example: 1A1 or 2B5); 2) enter valid fiche coordinate (example 1A or D5) and 3) use the commands below to change fiche coordinates.

> - (right arrow) moves one frame to right
< - (left arrow) moves one frame to left
^ - (caret) moves one frame down
- - (dash) moves one frame up

FIGURE CAPTIONS

Figure 2.1 - Schematic illustration of the philosophy behind BIRP searches. Two scratch disk files keep track of which pictures have met search constraints. The files, or bit maps, are initially set with all bits equal to unity. Searches are done on a single engineering parameter at a time. For instance, the user could input a value or set of values for the target (Mars, Phobos, etc.) parameter. Those data entries that do not have the appropriate entries have their bits in the bit map set to zero. During next search, on slant range, for instance, those entries with corresponding zero bits are skipped during the search. The old bitmap keeps the results of the previous search so that the user can back-up one search step, if needed.

Figure 7.1 - Schematic illustration of the print capabilities of BIRP. The print list consists of the pictures that have fulfilled search constraints, augmented with a user-selected list of engineering parameters to be printed out for each picture entry. The print list can be thought of as lying on a cylinder, with the first and last picture entry being adjacent to one another. The commands ADD, COPY, and CLEAR modify the list of engineering parameters to be printed. The commands PRINT, NEXT, PREVIOUS, DUMP, and DELETE are used to control printing the picture list.


SEARCHES
DATA FILES
SUBSOLAR LONGITUDE
CENTER LATITUDE
SLANT RANGE
TARGET
OLD BIT MAP
CURRENT BIT MAP
PICTURE
1
N

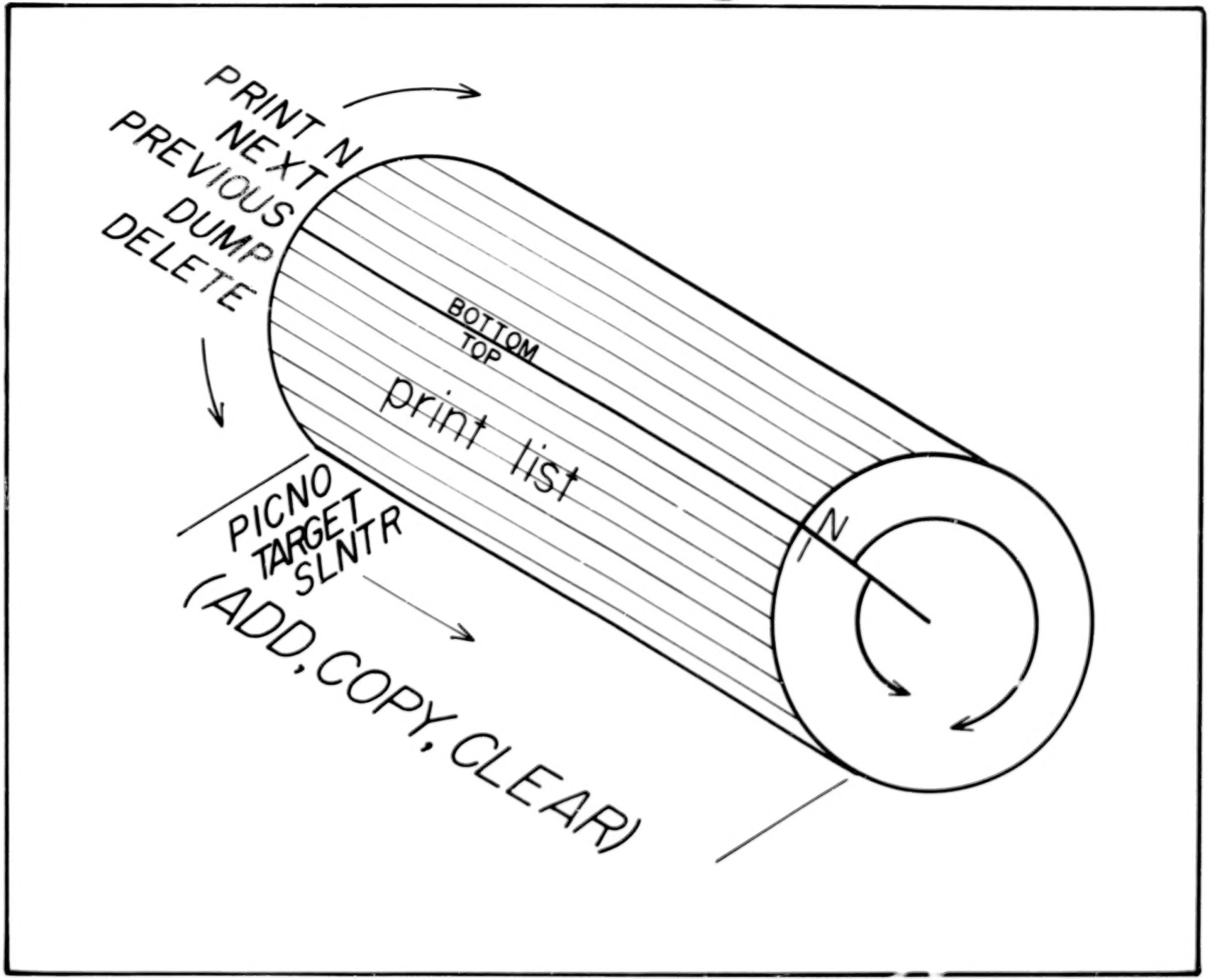

PRINT N
NEXT
PREVIOUS
DUMP
DELETE
BOTTOM
TOP
print list
N
PICNO
TARGET
SLNTR
(ADD, COPY, CLEAR)

Table 3.1
Header Area of PARAMETER.BRP Files

The numbers on the vertical axis are the absolute byte locations within the header area. The numbers on the horizontal axis are the relative byte offset from the byte locations shown on the vertical axis..skip 1

BYTE No.	0	1	2	3
0	MSB [Length of .BRP file: I*4 bytes]			LSB
4	MSB [Total Number of pictures: I*4]			LSB
8	MSB [Relative start byte of description area]			LSB
12	MSB [Relative start byte of data area: I*4]			LSB
16	Width/Ext.info	I/O Conv. type	Scan code	[1st letter]
20	Format Conversion Text (for data output)			
24	[last letter]	No. of Keys	Key 1 length	
28	Key 1 (example of 6 byte key, may be 8 bytes max)			
32		Key 2 length	Key 2 (example of 2 byte key)	
36	Length Des.Lin1			
40	Description line 1(example of 11 byte line, may be up to 80)			
44				
48	Length Des.Lin2	Description line 2		
52	(example of 7 byte line. Note: Max. total 32767 char.(bytes))			
54	Start of Data Area: 'Width-code' bytes per record (one/picture)			
58	MSB byte always first. If bit-type data (width-code= 0,1), highest bit of each byte is first 'record'.			

Note: If I/O type = 0,3,5 or 6 the keys section is omitted.
The structure below is used if I/O type=4 (real scale factor).

24		MSB		
28		Real *8 Scale Factor		
32		Des.Line 1 Lngh	Description Line 1	

Table 3.2
Index Buffer Structure

The numbers on the vertical axis are the absolute byte location within the index buffer. The numbers on the horizontal axis are the relative byte offsets from the byte locations shown on the vertical axis.

BYTE No.	0	1	2	3
0	LSB [Relative start of description area (I*4 bytes)]			MSB
4	LSB [Relative start of data area (I*4 bytes)]			MSB
8	Width Code(1-7)	Width (0=bit)	Extra Info.	I/O Conv.Type
12	Scan Code	(1st Letter)	Format Conversion Text	
16		(Last Letter)	No. of Keys	Key 1 length
20	Key 1 (example of 5 byte key, may be 8 bytes max.)			
24		Key 2 length	Key 2 (example of 2 byte key)	
28	Length Des.Lin1			
36	Description line 1(example of 11 byte line, may be up to 80)			
36				
40	Length Des.Lin2	Description line 2		
44	(example of 7 byte line. Note: Max. total 32767 char.(bytes))			
48	Start of Data Area: 'Width-code' bytes per record (one/picture)			
52	MSB byte always first. If bit-type data (width-code= 0,1), highest bit of each byte is first 'record'.			

Note: 1) Byte 9 contains the number of bytes in each data record.
Byte 9 is 0 for bit data.
2) If I/O type = 0,3,5 or 6 the keys section is omitted.
The structure below is used if I/O type=4 (real scale factor).

16	MSB			
20	Real *8 Scale Factor			
24	Des.Line 1 Lngh	Description Line 1		

Table 3.3
The Scan Control Table

The numbers on the vertical axis are the absolute byte locations within the scan control table. The numbers on the horizontal axis are the relative byte offests from the byte locations shown on the vertical axis.

BYTE No.	0	1	2	3
0	LSB [Start of data area in Parameter.Brp file (I*4 bytes)]			MSB
4	Scan Code	Width Code	No. of Values/Ranges*2 entered	
8	Log2(Widthcode)	[Range/Value data entered max. 512 bytes total]		

Table 3.4
BIRP.BRP File for Viking Orbiter

```
 39757,PICTURES
VIKING ORBITERS 1 AND 2
FRAMES 003A01 TO 974A10 AND 004B01 TO 705B52 ARE INCLUDED IN THIS LIBRARY.
TYPE "HELP" IF YOU'RE NEW TO BIRP.
PARAMETERS
----------
BOXNO        10 DEGREE BOX NUMBER CONTAINING CENTER OF PICTURE.      [RANGE: 1 - 648]
CAMERA       CAMERA USED: WHICH CAMERA OF ORBITER TOOK PICTURE.      [VALUES: A OR B]
CLAT LATITUDE OF PICTURE CENTER: +90=N.POLE,-90=S.POLE [RANGE:-90.0 TO +90.0]
CLONG        WEST LONGITUDE OF PICTURE CENTER:                       [RANGE:0-360.0]
CRAFT        SPACECRAFT: WHICH VIKING ORBITER.                       [VALUES:1 OR 2]
EMANG        EMISSION ANGLE: NORMAL TO SURFACE = 0 DEGREES           [RANGE:0-90.0]
FILTER       FILTER USED [VALUES: BLUE, MINUS (-BLUE), VIOLET, CLEAR, GREEN,OR RED]
INANG        INCIDENCE ANGLE: NORMAL TO SURFACE = 0 DEGREES          [RANGE:0-90.0]
LATLON       AREA SEARCH FILE: NON PRINTABLE
LS   AEROCENTRIC LONGITUDE OF THE SUN: POSITION OF SUN       [RANGE:0-360.0]
MCNO MARS CHART NUMBER:MARS MAPPING CHART (USGS QUADRANGLE) [VALUES:1 TO 30]
PHANG        PHASE ANGLE:ANGLE BETWEEN EMISSION AND INCIDENCE ANGLES [RANGE:0-180.0]
PICHT        PICTURE HEIGHT: HEIGHT OF PICTURE FRAME IN KILOMETERS.  [RANGE:2-2000]
PICNO        PICTURE SEQUENCE  : REVOLUTION, SPACECRAFT, PICTURE COUNT IN ORBIT.
     STANDARD PICTURE SEQUENCE NUMBER: XXXYZZ, WHERE XXX=ORBIT OR REVOLUTION
     NUMBER, Y=SPACECRAFT(A=VIKING 1, B=VIKING 2, S=VIKING 1 SURVEY MISSION)
     ZZ=PICT COUNT WITHIN ORBIT
PICWD        PICTURE WIDTH: WIDTH OF FRAME IN KILOMETERS.            [RANGE:2-2000]
QUAL PICT. QUALITY (DEGREE OF HAZINESS) [VALUES:CLEAR,SLIGHT,MODERATE,DENSE]
RES  RESOLUTION: SIZE OF A PICTURE ELEMENT (PIXEL) IN METERS.[RANGE:1-1500]
REVNO        REVOLUTION NUMBER: ORBIT NUMBER OF SPACECRAFT           [RANGE:0-999]
SLANTR       SLANT RANGE: VO TO PICT. CENTER ON MARS/SATELLITE (KM).[RANGE:250-30000]
TARGET       SUBJECT.  [VALUES: PHOBOS, DEIMOS, MARS, STAR, TERM(TERMINATOR) OR LIMB]
```

Table 4.1

BIRP MEMORY OVERLAY STRUCTURE

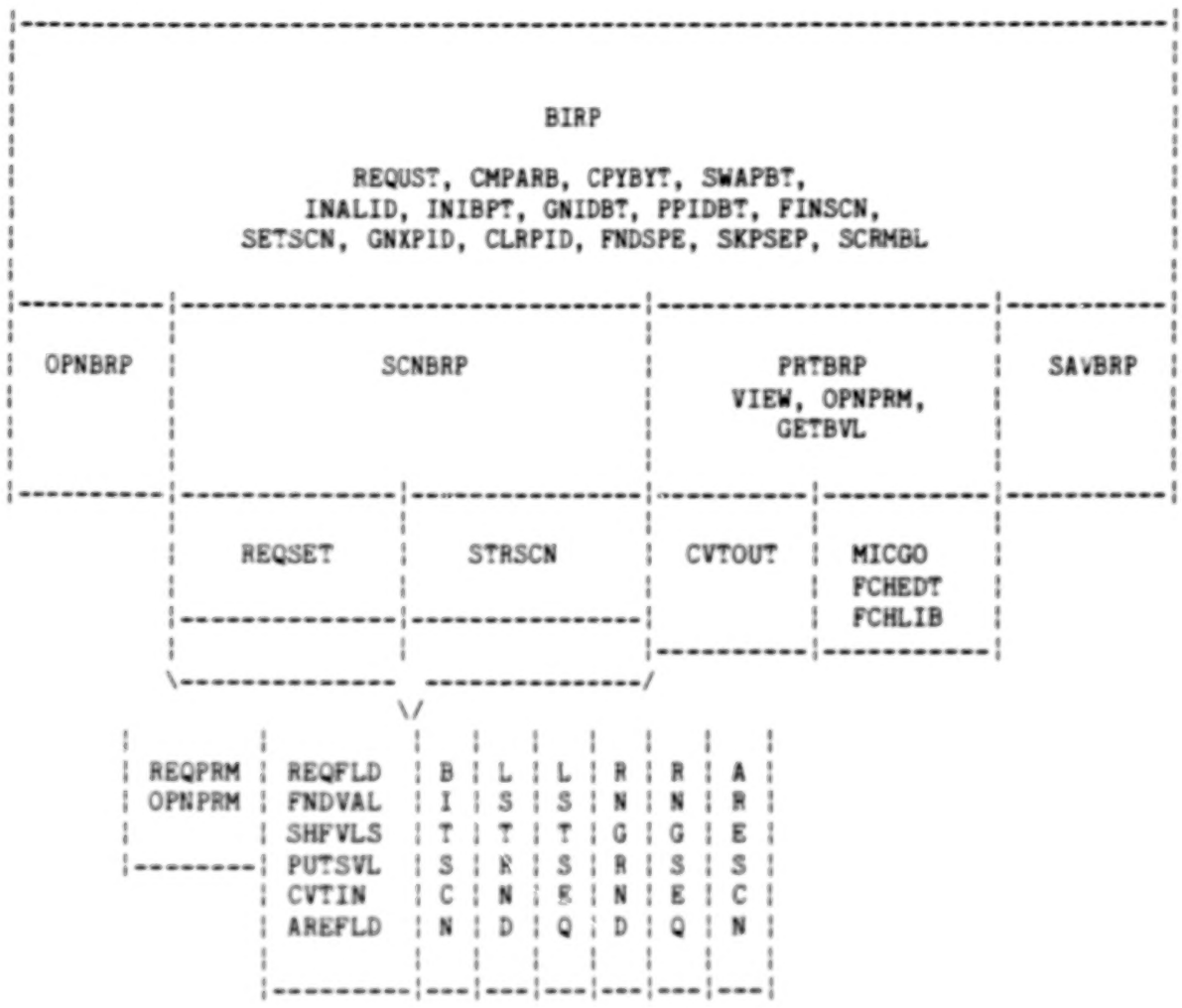


Table 4.1. BIRP is heavily overlayed so that it may run in the limited memory of a typical minicomputer. The overlay structure saves memory by reading routines into memory only as they are needed and simultaneously displacing routines that are no longer needed. No two routines at the same level in the chart can be in the memory at once. The segment at the top of the chart containing BIRP is called the root. The root is always resident in memory. The names refer to the various BIRP routines. Note that the ARESCN branch also includes the routines BOX, QUAD and NTRGLE.

TABLE 6.1

INPUT REQUESTED BY REQFLD

SCAN CODE	DATA TYPE	INPUT REQUIRED
0	0	Y: CRITERIA PRESENT, N: ABSENT
0	1	[TEXT] ONLY TWO POSSIBLE VALUES
1,2	2,3,5	[VALUE] TEXT OR NUMBER
3,4	3,4,6	[RANGE] PAIR OF NUMBERS
5,6	3,4,6	LATITUDE OR LONGITUDE OF POINT

Table 6.2
BITWISE SCAN OF A DATA FILE

	PICTURE ID	1 2 3 4 5 6 7 8
	PARAMETER VALUES	1 0 0 1 1 1 0 1
	USER INPUT	0 0 0 0 0 0 0 0
STEP 1:	"EXCLUSIVE OR" RESULT	1 0 0 1 1 1 0 1
	BIT MAP	1 0 1 0 0 1 1 1
STEP 2:	LOGICAL "AND" RESULT	1 0 0 0 0 1 0 1

Table 6.2. The purpose of this example is to illustrate the bitwise scanning method. The user, in this example, is searching for pictures with a value of 1 (i.e., Viking Orbiter A). The result of the logical "and" replaces the old value of the bit map. In this example pictures 1, 6 and 8 remain selected, while pictures 3 and 7 will no longer be available for subsequent searches. Note that the user has already performed at least one search prior to this example, and that pictures 2, 4 and 5 have been eliminated by the previous search(es). As a result, this example leaves pictures 2, 4 and 5 eliminated.

Table 6.3
SORTED LIST SCAN

SCAN CONTROL TABLE					5				11	12		16				
PICTURE VALUES	1	2	3	4	5	5	6	7	10	12	14	16	18	18	19	20
NEW BIT MAP	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	0
	¦----------¦												¦----------¦			
	AUTOMATIC <												AUTOMATIC >			

Table 6.3. A sorted list search is illustrated in this example, but it also applies to the sorted range search. The entries in the scan control table are 5, 11, 12 and 16. "AUTOMATIC <" means that the values for those pictures are automatically less than all of the entries in the scan control table because they are less than the first entry. In a similar manner, "AUTOMATIC >" means that the values for those pictures are automatically greater than all of the entries in the scan control table. Of course, a value of 1 in the bit map means that a picture passed the search test, while a value of 0 means that a picture failed the search test.

Table 11.1

Data Structure of PNAME.IDX Files

LINE No.	Contents
1	I7 integer: Number of pictures or data records. Must be same for all files in a library.
2	I1 integer: Width code of each datum.
3	I2 integer: Extended information code.
4	I3 integer: Data I/O type code.
5	I1 integer: Scan code.
6	A6 text: Output conversion format.
7 on....	Depends on data I/O code: 1,2: Text Keys 4: Real scale factor as F16.0.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-3299		

4. Title and Subtitle	5. Report Date
BIRP: Software for Interactive Search and Retrieval of Image Engineering Data	July 1980
	6. Performing Organization Code SL-4
7. Author(s) Raymond E. Arvidson, Lawrence K. Bolef, Edward A. Guinness, and Peter Norberg	8. Performing Organization Report No.
	10. Work Unit No.
9. Performing Organization Name and Address McDonnell Center for the Space Sciences Department of Earth and Planetary Sciences Washington University, St. Louis, MO 63130	11. Contract or Grant No. NSG-7087
	13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546	14. Sponsoring Agency Code

15. Supplementary Notes

NASA Headquarters Technical Monitor: Joseph Boyce
Final Report

16. Abstract

BIRP (Better Image Retrieval Programs) is a set of programs to interactively sort through and to display a database, such as engineering data for images acquired by spacecraft. This document provides an overview of the philosophy of BIRP design, the structure of BIRP data files, and examples that illustrate the capabilities of the software.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Image Engineering Data Interactive Search and Retrieval	Unclassified-Unlimited Subject Category 61

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	71	$5.25

For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1980

END

5-22-81

90%